차 례

머 리 말

위대한 령도자 **김정일**원수님께서는 다음과 같이 지적하시였다.

《프로그람을 개발하는데서 기본은 우리 식의 프로그람을 개발하는것입니다. 우리는 우리 식의 프로그람을 개발하는 방향으로 나가야 합니다.》

프로그람기술을 빨리 발전시키는것은 나라의 정보기술을 높은 수준에 올려세우기 위한 중요한 요구의 하나이다.

위대한 령도자 **김정일**원수님의 현명한 령도에 의하여 오늘 우리 나라에는 정보기술을 빨리 발전시킬수 있는 물질기술적토대가 튼튼히 마련되였으며 새로운 정보기술성과들이 련이어 이룩되고있다.

나라의 정보기술을 높은 수준에 올려세우는데서 중요한것은 우리 식 조작체계 《붉은별》에서 응용할수 있는 프로그람들을 빨리 발전시키는것이다.

5학년 《콤퓨터》에서는 대표적인 프로그람작성언어인 C를 리용하여 프로그람작성의 기초적인 방법들을 배운다.

프로그람을 작성한다는것은 풀려는 문제를 잘 파악해서 그 풀이순서와 방법을 결정한 다음 그것을 콤퓨터가 아는 프로그람언어로 서술해주는것을 말한다.

먼저 프로그람작성에서 기초로 되는 알고리듬과 프로그람언어에 대한 기본지식들에 대하여 배운다. 그리고 간단한 프로그람작성과정을 통하여 C언어의 기본구성요소들에 대하여 배우게 된다.

모든것은 기초가 든든해야 훌륭한 결실을 볼수 있다.

우리는 이 과목학습을 실속있게 하여 앞으로 보다 능률적인 우리 식의 응용프로그람들을 개발할수 있는 기초를 튼튼히 다짐으로써 우리 학생들이 사회주의강성대국의 밝은 앞날을 떠메고나갈 훌륭한 과학기술인재로 자라나도록 크나큰 사랑과 은정을 베풀어주고계시는 위대한 령도자 **김정일**원수님의 믿음과 기대에 높은 실력으로 보답하여야 한다.

제1장. 프로그람작성의 기초지식

콤퓨터의 동작원리를 다 파악하려면 콤퓨터장치의 요소요소가 다 어떻게 맞물려 동작하는가를 잘 알아야 할것이다. 이에 못지 않게 프로그람작성원리를 잘 알아야 콤퓨터의 동작원리를 잘 파악할수 있으며 콤퓨터를 더 잘 다룰수 있다.

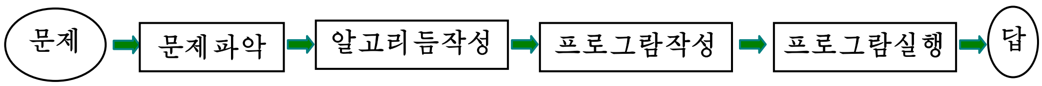
제1절. 콤퓨터로 문제를 풀려면

1. 문제풀이단계

례를 들어 어떤 수학문제를 푼다고 할 때 먼저 그 문제를 여러번 읽으면서 문제의 요구를 정확히 파악한데 기초하여 풀이방법을 찾아내야 한다. 그다음 찾아낸 풀이방법대로 종이우에 차례차례 풀어서 주어진 문제의 답을 얻는다. 즉 문제풀이과정은 다음과 같다.



이 문제를 콤퓨터로 풀려고 하는 경우에는 다음과 같은 과정을 거쳐야 한다.



콤퓨터에 의한 문제풀이과정을 단계별로 보기로 하자.

1) 문제파악

이 단계에서는 문제의 요구를 정확히 파악하여 문제풀이에 필요한 조건으로 되는 **입력자료**(또는 초기자료)와 구하려는 **출력자료**(결과자료)를 찾는다.

풀이를 위한 수학적인 정식화(수학적모형화)를 진행한다.

례 1: 두 밑변과 높이에 의하여 제형의 면적을 계산하는 문제를 보기로 하자.

입력자료: 두 밑변 a, b

높이 h

출력자료: 제형의 면적 S

수학적모형화: $S=\frac{1}{2}(a+b)h$

수학적모형화가 이루어지면 수값풀이를 진행하여야 한다.

례 2: 2차방정식 $ax^2 + bx + c = 0$의 실수풀이를 구하자.

입력자료: 2차방정식의 곁수 a, b, c

출력자료: 실수풀이 x

수학적모형화: 판별식 $d = b^2 - 4ac$의 값에 따라

$d > 0$인 경우 $x_{1,2} = \frac{-b \pm \sqrt{d}}{2a}$

$d = 0$인 경우 $x = \frac{-b}{2a}$

$d < 0$인 경우 풀이가 없다.

수학적모형화의 매 경우에 따르는 수값풀이를 진행한다.

2) 알고리듬작성

알고리듬이란 주어진 문제를 풀기 위한 과정 즉 어떤 연산들을 어떤 순서로 집행하겠는가를 밝혀놓은것을 말한다.

문제파악이 끝나면 풀어야 할 처리내용을 자기가 알수 있게 순서대로 적어놓아야 한다. 즉 알고리듬을 작성하여야 한다.

앞의 례들을 다음과 같이 순서화할수 있다.

례 1: ① 변 a, b와 높이 h를 입력한다.

② 제형의 면적 $S=\frac{1}{2}(a+b)h$를 구한다.

③ S를 출력한다.

례 2: ① 2차방정식의 결수 a, b, c를 입력한다.

② 판별식 $d=b^2 - 4ac$를 구한다.

③ d에 따라 풀이를 구한다.

④ 풀이를 출력한다.

알고리듬작성과정은 콤퓨터에 의한 문제풀이단계에서 가장 기본적인 부분을 이루며 어렵고 창조적인 사고활동을 요구하는 작업과정이다. 구체적인 알고리듬에 대하여서는 뒤에서 보기로 한다.

3) 프로그람작성

알고리듬을 콤퓨터가 리해할수 있는 언어로 바꾸어놓는 과정을 프로그람작성이라고 한다. 앞의 제형면적계산례를 C언어프로그람으로 작성하면 다음과 같다.

```
#include <stdio.h>
main()
{
    int a, b;                 // 옹근수형변수 a, b선언
    int h;                    // 옹근수형변수 h선언
    float S;                  // 실수형변수 S선언
    scanf("%d%d%d", &a,&b,&h);    // 자료입력
    S=1/2*(a+b)*h;                // 면적계산
    printf("%f\n", S);            // 면적출력
}
```

4) 프로그람검열

프로그람검열은 앞단계들의 검토과정이다.

매 단계에서 여러가지 잘못 즉 오유가 있을수 있다. 오유에는 두가지가 있다. 하나는 프로그람언어의 문법적오유이며 다른 하나는 잘못된 사고로 인한 론리적오유이다. 문제파악단계와 알고리듬작성단계에서는 론리적오유가 발생될수 있으며 프로그람작성단계에서는 문법적오유가 나올수 있다.

그러므로 새로 작성한 프로그람을 실행하기 전에 검열단계를 거쳐야 한다.

검열과정에 모든 오유들이 발견되고 수정되여야 한다.

5) 프로그람실행

프로그람개발환경을 마련하고 앞에서 작성한 프로그람을 건반을 통해 편집한 다음 각종 지령을 주어 실행시켜야 한다.

실행에서 오유가 나오면 수정하여 다시 실행시킨다. 오유가 없는 경우에는 답을 얻게 된다.

2. 알고리듬작성

1) 알고리듬의 분류

문제를 빠른 시간에 정확하게 풀자면 그 문제를 풀기 위한 단계를 옳바로 결정하여야 한다.

콤퓨터로 풀어야 할 문제가 제기되면 콤퓨터가 풀어야 할 문제풀이순서인 알고리듬을 작성한 다음 이것을 콤퓨터가 알수 있는 언어로 바꾸어 콤퓨터에 입력시키면 콤퓨터는 그것에 의하여 동작을 수행하게 된다.

같은 문제라도 그것을 어떻게 구성하는가에 따라 서로 다른 알고리듬이 될수 있다. 례로 두 수의 최대공통약수를 구하는 알고리듬에는 두 수가 공통으로 가지는 약수들가운데서 제일 큰것을 찾는 방법과 유클리드련제법을 리용하여 구하는 방법이 있다.

어느 방법을 선택하는가에 따라 연산순서 즉 알고리듬이 달라진다.

이와 같이 여러가지 방법가운데서 어느것을 선택하겠는가 하는것은 콤퓨터의 리용측면을 놓고 결정하여야 한다. 좋은 알고리듬이라고 할 때는 연산조작회수가 적고 프로그람의 크기가 작은것이다.

- 직선형(순차형)구조

다음과 같이 두 밑변과 높이가 주어졌을 때 제형의 면적을 구하는 알고리듬을 보기로 하자.

① 시작

② 변수 a, b에 두 밑변의 길이값을, 변수 h에 높이값을 넣는다.

③ 변수 s에 $\frac{1}{2}(a+b)h$를 넣는다.

④ 변수 s에 있는 결과값을 화면에 표시한다.

⑤ 끝

여기서 변수는 수학문제를 풀 때의 변수가 아니라 프로그람에서의 변수 즉 콤퓨터

의 기억장소를 의미한다. 그러므로 이제부터 우리가 알고리듬에서 쓰게 되는 모든 변수는 기억장소라고 생각하고 콤퓨터에서 진행되는 과정을 생각하면서 콤퓨터에 지령을 주듯이 알고리듬을 작성해주어야 한다.

시작을 하면 콤퓨터는 새로운 과제를 수행할 준비단계에 들어간다.

콤퓨터가 두 밑변의 길이와 높이값을 알아야 면적을 계산하므로 콤퓨터에 이 3개의 값을 넣어주어야 하는데 콤퓨터는 모든 자료를 기억기에 기억하므로 기억장소 a, b에는 두 밑변의 길이값을, h에는 높이값을 넣어주어야 한다.

기억장소 s에는 면적값을 계산하여 넣어주고 이것이 화면에 표시되여야 사용자가 알게 되므로 기억장소 s에 있는 면적값을 화면에 출력해야 한다.

처음에 주어지는 값들을 변수 a, b, h에 넣어주는것을 입력이라고 하며 계산한 값을 변수 s에 넣어주는것을 값주기라고 하고 기호 ←로 표시한다.

우의 알고리듬은 처음부터 씌여진 순서대로 처리하게 구성되여있다. 이런 형태의 알고리듬구조를 직선형구조라고 한다.

- 갈래형구조

한 학생의 평균성적을 입력하여 3.5이상이면 《합격》, 그아래이면 《불합격》이라고 표시하는 알고리듬을 만들자.

① 시작
② s(성적)초기값입력
③ 만일 s≧3.5 이면 ④에로 이행, 그렇지 않으면 ⑥에로 이행
④ 《합격》출력
⑤ ⑦에로 이행
⑥ 《불합격》출력
⑦ 끝

이 알고리듬은 처리가 번호순서로 차례로 진행되는것이 아니라 도중에 조건에 따라 두갈래가운데서 어느 한 갈래를 따라 처리하게 되여있다. 즉 입력하는 성적값이 3.5이상이면 ①, ②, ③, ④, ⑤, ⑦의 순서로, 3.5보다 아래이면 ①, ②, ③, ⑥, ⑦의 순서로 처리가 진행된다. 이와 같이 조건에 따라 여러 갈래가운데서 어느 한 갈래를 택하여 처리를 진행하는 형태의 알고리듬을 갈래형구조라고 한다.

- 순환형(반복형)구조

수렬의 합 s = 1 + 2 + 3 + 4 + … + 100을 계산하는 알고리듬을 만들어보자.

이것을 계산하기 위해 1씩 주기적으로 증가하는 수를 변수 i에 넣고 현재의 i값까지의 합을 변수 s에 넣는것으로 하고 표를 통하여 그 과정을 보자.

i계산	i값	s계산	s값
i←0	0	s←0	0
i←i+1	1	s←s+i	1
i←i+1	2	s←s+i	1+2
…	…	…	…
i←i+1	100	s←s+i	1+2+3+…+100

표에서 보는바와 같이 첫 단계 i←0, s←0를 제외하고는 i←i+1, s←s+i라는 계산을

100번이나 반복한다.

여기서 i←i+1은 이미 i기억장소에 기억된 값에 1을 더하여 다시 i기억장소에 기억시킨다는 뜻을 담고있다. 그러면 i에는 1씩 증가된 값이 넣어질것이다.

s←s+i는 이미 얻어진 합 s에 이제 새로 들어온 i값을 더하여 다시 변수 s에 넣으므로 s에는 i가 변할 때마다 그 값까지의 합이 설정된다.

그러므로 i←i+1, s←s+i를 100번 반복수행한 다음 결과를 출력하도록 알고리듬을 구성하면 된다. 몇번 반복수행하는가는 매번 i의 값이 얼마인가를 알아보고 100이 되면 반복을 그만하게 하면 된다.

알고리듬은 다음과 같다.

① 시작
② i←0 , s←0
③ i=100이면 ⑦에로 이행, 그렇지 않으면 다음처리(④에로)
④ i←i+1
⑤ s←s+i
⑥ ③에로 이행
⑦ s값 출력
⑧ 끝

우의 알고리듬에서 반복처리를 하게 하는 ③과 ⑥을 변경하여 다음과 같이 알고리듬을 만들수 있다.

① 시작
② i←0, s←0
③ i←i+1
④ s←s+i
⑤ i<100이면 ③에로 이행, 그렇지 않으면 다음처리(⑥에로)
⑥ s값 출력
⑦ 끝

이와 같이 알고리듬구성은 여러가지로 작성할수 있는데 보다 간단할수록(처리회수가 작을수록) 더 좋은 알고리듬으로 된다.

우의 알고리듬은 어떤 조건을 만족할 때까지(i값이 100이 될 때까지) 처리를 반복(③, ④, ⑤부분)하게 되여있다. 이런 형태를 순환형구조, 반복처리하는 부분을 순환부라고 한다.

일반적으로 알고리듬은 직선형, 갈래형, 순환형 또는 그것들의 혼합으로 이루어져있다.

알고리듬은 다음의 조건을 만족시켜야 한다.

① 같은 자료에 대하여 실행하면 같은 결과가 얻어져야 한다.
② 처리는 반드시 유한번의 처리로 끝나야 한다.
③ 하나의 문제에서만이 아니라 그와 같은 형태의 여러 문제에서도 모두 적용할수 있어야 한다.

작성자에 따라 알고리듬이 달라질수 있다.

2) 알고리듬의 표시방법

알고리듬은 사용자가 콤퓨터가 풀어야 할 문제의 처리내용을 쓴것이므로 그것을 작성하지 않은 사람도 보고 쉽게 프로그람으로 옮길수 있게 알아보기 편리하게 되여야 한다. 이로부터 알고리듬을 표시하는 방법에는 글로 서술하는 방법, 그림으로 표시하는 방법 등 여러가지 방법이 있다.

글로 서술하는 방법에는 문제풀이순서를 사람들이 사용하는 일반언어로 서술하는 방법(우의 례들)과 프로그람모조코드(의사코드)로 표시하는 방법이 있다.

그림으로 표시하는 방법에서는 문제풀이의 전 과정을 직관성있게 나타내기 위하여 흔히 흐름도나 블로크도와 같은 그림으로 표시한다.

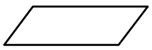
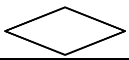
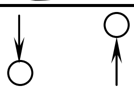
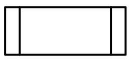
그러면 직관적효과가 강한 흐름도와 블로크도에 대하여 보기로 하자.

- 흐름도

몇개의 기하학적도형과 처리의 방향을 나타내는 화살을 리용하여 풀이의 전 과정을 나타내는 알고리듬을 직관적으로 표시한것을 흐름도라고 한다.

흐름도는 프로그람작성에 매우 편리하며 직관성이 있으므로 프로그람에서 나타나는 오유를 쉽게 찾아낼수 있다.

흐름도에서 리용되는 기하학적도형들은 다음과 같다.

기호이름	기 호	의 미
시작 또는 끝		문제풀이의 시작 또는 끝을 나타낸다.
자료입력		문제풀이에 필요한 초기자료를 넣는다.
처리기호		하나 또는 여러개의 처리내용을 표시한다.
판단기호		비교, 판단되는 내용을 표시한다.
출력기호		출력되는 자료를 표시한다.
결합기호		흐름선이 이어진다는것을 표시한다.
함수호출		함수를 호출한다.
흐름선		문제풀이흐름방향을 표시한다.

① 직선형구조

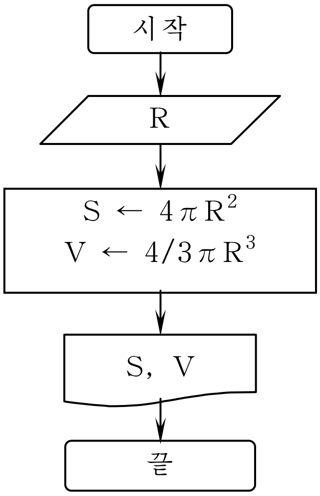
구의 겉면적과 체적을 구하는 알고리듬을 흐름도로 작성하자.

입력자료: 구의 반경 R

출력자료: 겉면적 S, 체적 V

수학적모형화: $S = 4\pi R^2$, $V = 4/3\pi R^3$

흐름도:



② 순환형구조

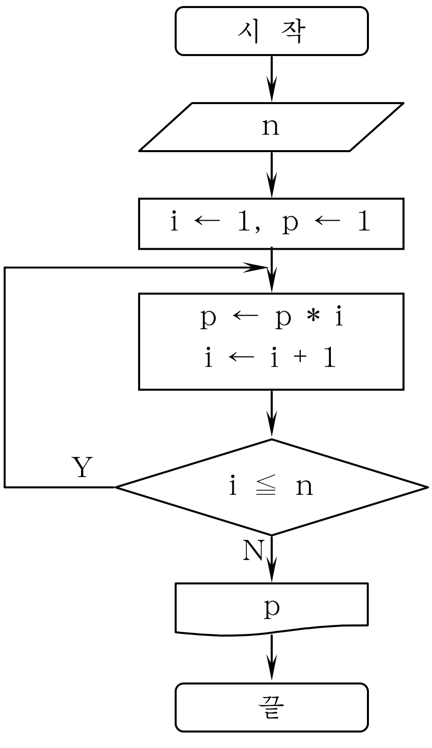
n차례곱을 구하는 알고리듬을 흐름도로 작성하자.

입력자료: 차례곱한계 n

출력자료: 차례곱 p

수학적모형화: p = Πi

흐름도:



③ 갈래형구조

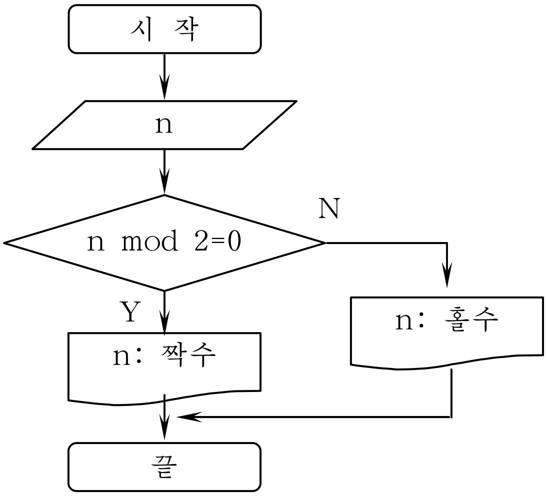
정의 옹근수 n을 읽고 그것의 짝홀성을 판정하는 알고리듬을 흐름도로 작성하자.

입력자료: 정의 옹근수 n

출력자료: n의 짝홀성

수학적모형화: n을 2로 나눈 나머지(이것을 프로그람에서는 n mod 2로 표시한다.)가 0이면 짝수, 1이면 홀수

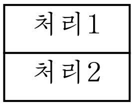
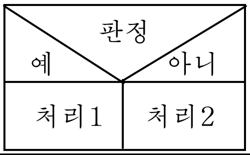
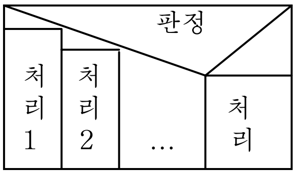
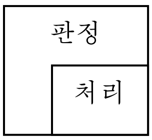
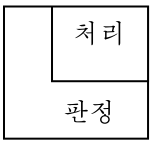
흐름도:



- **블로크도**

블로크도는 흐름도와 같이 조종의 전달을 따로 밝히지 않고 포함구조만으로 표현하는 알고리듬의 표현방법의 하나이다. 이것을 **NS도식**이라고도 한다.

기호들의 의미는 다음과 같다.

구분	NS도식	의 미
순차구조		순차적으로 처리되는 내용표시
선택구조		두개의 처리가운데서 조건판정에 따라 어느 하나의 처리를 진행
		여러개의 처리가운데서 조건판정에 따라 어느 하나의 처리를 진행
반복구조		먼저 조건을 판정하고 그에 따라 처리를 반복
		뒤에서 조건을 판정하고 그에 따라 처리를 반복

① 직선형구조

반경이 R인 원에서 중심각이 120° 일 때 활등의 길이를 구하는 알고리듬을 블로크도로 작성하자.

입력자료: 반경 R, 중심각 α = 120°

출력자료: 활등의 길이 ℓ

수학적모형화: $\ell = \pi/180 * \alpha * R$

블로크도:

R입력, $\alpha \leftarrow 120$
$\ell \leftarrow \pi/180 * \alpha * R$
ℓ출력

② 직선형구조

두 수 a, b의 값이 각각 3, 4일 때 두 수의 합을 계산하는 알고리듬을 블로크도로 작성하자.

입력자료: 두 수 a=3, b=4

출력자료: 합 S

수학적모형화: S = a + b

블로크도:

a ← 3, b ← 4
S ← a + b
S출력

③ 갈래형과 순환형의 혼합구조

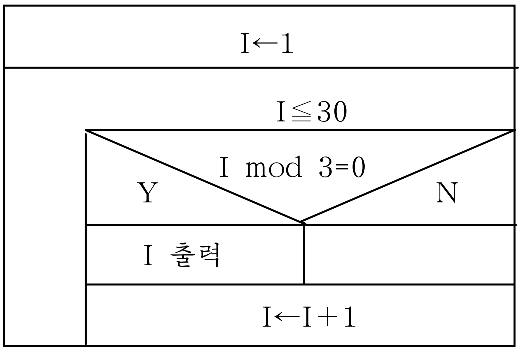
30보다 작은 3의 배수를 구하는 알고리듬을 블로크도로 작성하자.

입력자료: 자연수한계 30

출력자료: 3의 배수들의 모임

수학적모형화: x mod 3 = 0으로 되는 모든 x

블로크도:



련습문제

1. 다음 문제들에서 입력자료와 출력자료를 찾고 수학적모형화를 진행하여라.
   1) 10명 학생들의 수학성적이 있다. 최우등생수, 우등생수를 계산하여라.
   2) 60보다 작은 6의 배수를 모두 말하여라.
   3) 우리 학급 학생들가운데서 제일 큰 학생의 키는 얼마인가?
   4) 우리 학급의 키측정자료를 큰 순서대로 배렬하여라.
   5) 30부터 50까지의 자연수에서 씨수를 골라내여라.
2. 다음 문제의 알고리듬을 작성하여라.
   1) 두 점의 자리표가 주어질 때 두 점사이의 거리를 구하여라.
   2) 웃반경, 아래반경, 높이가 주어질 때 원뿔대의 체적을 구하여라.
   3) 세 점의 자리표가 주어졌을 때 3각형의 면적을 구하여라.
   4) 한 변의 길이가 a인 바른3각형의 내접원의 면적과 둘레의 길이를 구하여라.
   5) 길이가 a, b, c인 세 선분이 3각형의 변을 이룰수 있는가를 판정하여 그 면적을 구하여라.
   6) 1부터 100까지 자연수들가운데서 짝수들의 개수를 구하여라.
   7) 2010년 4월 15일이 목요일이라는것을 알고 이달의 임의의 날자를 입력하면 그에 대한 요일을 출력하여라.
   8) 어떤 옹근수 x가 씨수인가, 합성수인가를 판정하여라.
   9) 1부터 100까지 자연수가운데서 씨수를 출력하여라.
   10) 24×24표를 계단별로 구하여라.




콤퓨터상식

알고리듬

알고리듬은 중앙아시아의 수학자 알리 호레즈미의 이름에서 유래되였다.

그가 9세기초에 쓴 수의 표기법과 산수연산규칙에 관한 책이 12세기에 라틴어로 번역되여 유럽에 전해졌는데 저자의 이름이 Algorithmi로 잘못 표기됨으로써 알고리듬이라는 용어가 생겨나게 되였다. 알고리듬이 잘 작성되여야 앞으로 콤퓨터에 주는 지령들이 잘 작성될수 있다. 따라서 콤퓨터가 문제를 잘 푸는가 못 푸는가 하는것은 알고리듬을 얼마나 잘 작성하는가 하는데 관계된다.

이 세상에는 어떤 문제에 대한 알고리듬을 찾기 위해 일생을 바친 과학자들도 있다. 세상에는 알고리듬이 존재하는 문제도 있고 존재하지 않는 문제도 있다. 또한 알고리듬에는 효률이 높은 알고리듬도 있고 효률이 높지 못한 알고리듬도 있다. 효률이 높은 알고리듬 즉 좋은 알고리듬은 기억기용량을 작게 쓰고 처리회수가 될수록 작은것이다. 처리회수가 작아야 콤퓨터에서 실행시간이 짧아지며 한 문제를 푸는데 쓰는 기억용량이 작아야 다중과제처리콤퓨터가 단번에 보다 많은 문제를 풀수 있다.

제2절. 프로그람언어란 무엇인가

1. 프로그람작성언어

1) 프로그람작성언어

프로그람을 작성하자면 그것을 서술하기 위한 프로그람작성언어가 필요하다.

사람들사이의 대화는 반드시 어떤 언어(조선어, 영어, 중어 등)를 가지고 진행된다. 사람들사이에 사용하는 이러한 언어를 **자연언어**라고 한다.

마찬가지로 콤퓨터에서도 각 장치들사이 그리고 사람과 콤퓨터사이에 서로 《말》을 주고받자면 언어가 있어야 한다. 바로 이 언어가 프로그람작성언어이다. 프로그람작성언어는 콤퓨터가 수행하는 일을 서술하기 위한 언어적인 표현수단이다.

2) 프로그람작성언어의 분류

프로그람작성언어는 크게 범용언어와 특수문제방향어로 분류한다.

(1) 범용언어

범용언어는 말그대로 프로그람작성에서 널리 쓰이고있는 언어로서 발전정도에 따라 저급언어와 고급언어로 나눈다.

① 저급언어

저급언어는 콤퓨터의 하드웨어에 얽매인 언어로서 **기계어**(machine language)와 **아쎔블리어**(assembly language)로 나눈다.

기계어는 콤퓨터의 각 장치들사이에 통하는 언어이다.

기계어로 작성한 프로그람(기계어프로그람)은 하드웨어가 직접 실행시킬수 있다.

기계어는 0과 1의 약속된 수자렬로 구성되여있으며 하드웨어를 알아야만 프로그람을 작성할수 있기때문에 프로그람을 작성하고 다루는데서 까다롭고 품이 많이 든다.

기계어의 부족점을 덜기 위하여 명령이나 기억주소를 어떤 영어문자로 바꾸어 리용하는 언어가 아쎔블리어이다.

② 고급언어

고급언어는 콤퓨터에 얽매이지 않으며 자연언어에 가까운 언어로서 프로그람을 작성하기가 편리한 언어이다. 고급언어는 그의 구성에 따라 **수속형언어**와 **비수속형언어**로 나눈다.

- 수속형언어(procedure oriented language)

수속형언어는 콤퓨터가 해야 할 일을 순서대로 표시하는 언어이다. 따라서 이 언어로 작성한 프로그람은 시작과 끝이 있으며 시작에서 끝까지 집행과정을 1렬로 줄지어놓을수 있다.

대표적인 언어로서 C, Basic, Pascal 등을 들수 있다.

- 비수속형언어(nonprocedural oriented language)

이 언어는 인간의 사유원리에 기초하여 어떤 사물현상에 대한 판단을 표현하는 언어이다. 여기에는 론리형언어, 함수형언어, 객체지향언어가 있다.

여러가지 언어를 리용하여 두 수 a와 b의 더하기를 표시해보자.

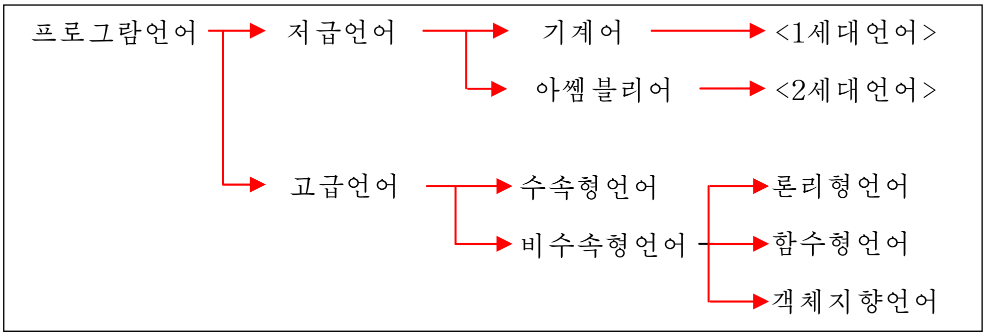
여러가지 언어를 리용한 두 수의 더하기

언어 / 조작	저급언어		고급언어 (C언어례)
	기계어	아쎔블리어	
a = 8	10111000 (조작코드) 0000000000001000 (값)	MOV AX,8	a = 8
b = 5	10111011 0000000000000101	MOV BX,5	b = 5
a + b	0000000111000111	ADD AX,BX	a + b

표에서 볼수 있는바와 같이 기계어는 1세대콤퓨터가 사용한 언어로서 콤퓨터가 직접 리해할수 있는 0과 1의 2진수로 명령을 표현하고있다. 그러므로 기계어로 작성한 프로그람을 실행할 때에는 번역할 필요가 없기때문에 실행속도가 빠르지만 콤퓨터의 종류에 따라 사용하는 명령이 다르므로 사람들이 리해하기 어렵다.

아쎔블리어는 기계어대신 그와 1:1로 대응되는 리해하기 쉬운 기호로 명령을 만든 언어이다. 그러므로 기계어보다는 좀 리해하기 쉽지만 콤퓨터내부의 등록기를 직접 조작하는것으로 하여 전문가가 아니면 역시 프로그람작성이 힘들다.

그러나 고급언어로 작성된 프로그람은 우리가 흔히 리용하고있는 수학적인 기호로 구성된것으로 하여 리해하기 쉽다.



(2) 특수문제방향어

특수문제방향어는 특별히 제한된 범위의 문제들을 처리하기 위한 전용언어이다.

2. C언어에 대한 개념

1) C언어의 특징

C언어는 현재 가장 많이 사용되고있는 프로그람작성언어이다.

C언어란 이름은 B언어 다음에 나온 언어라는데로부터 붙여진 이름이다.

초기에 C언어는 조작체계 Unix를 서술하기 위하여 1972년에 개발한 고수준프로

그람작성언어이다. Unix조작체계는 체계의 90%에 해당되는 부분을 C언어로 서술하였다. 그러므로 C언어를 체계개발언어라고도 한다. C언어는 체계개발을 위하여 만들어진 언어라는데로부터 Basic나 Pascal과 같은 고수준언어의 특징을 가지면서도 아쎔블리어로만 가능한 처리까지도 자유롭게 서술할수 있는 능력을 가지고있다.

C언어의 특징은 다음과 같다.

① 언어표현능력이 강하다.

C언어는 비트조작이나 주소관리 등 극히 세부적인 처리가 강하기때문에 아쎔블리어를 대신하여 여러가지 체계프로그람과 응용프로그람을 작성한다.

② 여러가지 형의 자료구조를 설계할수 있는 능력을 가지고있다.

③ C언어는 매우 간결하고 콤파일러가 작다.

다른 언어에서 Begin ~ End로 표시하는것을 { 와 }으로 , If a>b then ~ else를 3항연산자를 리용하여 간결하게 표시한다.

또한 프로그람실행과 관련한 자원도 적으며 기억공간도 적게 차지한다.

④ 언어가 만드는 코드의 질이 매우 높다.

C언어는 아쎔블리어에 대비할만큼 코드효률이 높다.

⑤ 이식성이 비교적 좋다.

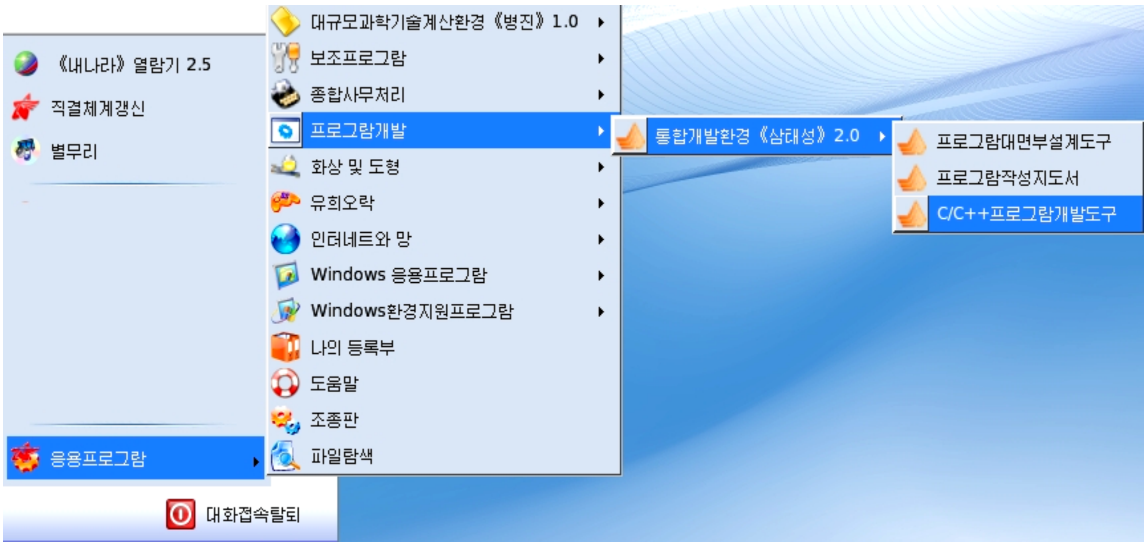
C언어로 작성한 프로그람은 수정하지 않거나 혹은 약간 수정하면 완전히 다른 환경에 옮겨 실현할수 있다. C언어의 결함은 연산자의 우선권종류가 너무 많으므로 기억하기 힘들며 자료형검사능력이 약하고 비교적 쉽게 절환시킬수 있으므로 안전성이 높지 못한것이다.

2) C프로그람의 구조

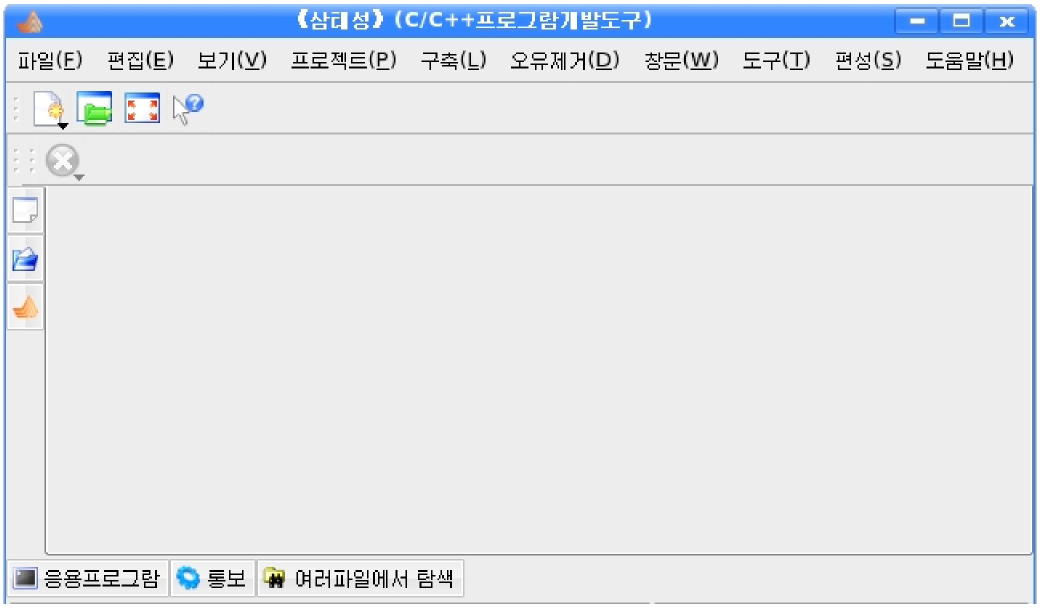
C언어로 프로그람을 작성하려면 C프로그람의 구조를 잘 알아야 한다.

례로 다음과 같이 통합개발환경 《삼태성》에서 주어지는 례프로그람을 통하여 그 구조를 알아보기로 하자.

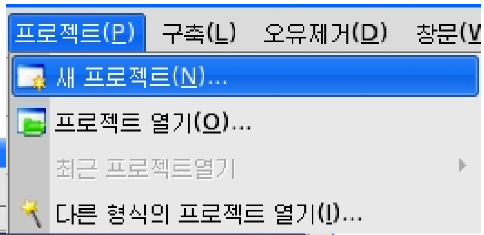
《시작》→《응용프로그람》→《프로그람개발》→《통합개발환경《삼태성》2.0》→《C/C++프로그람개발도구》를 실행한다.

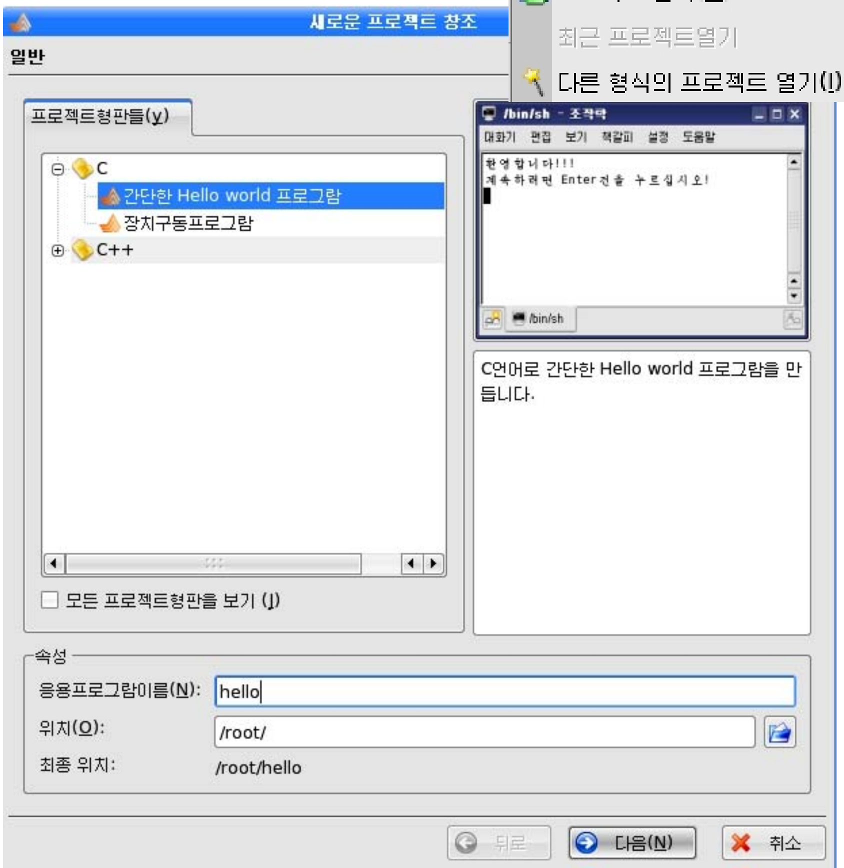


그러면 《삼태성》(C/C++프로그람개발도구)창문이 펼쳐진다.



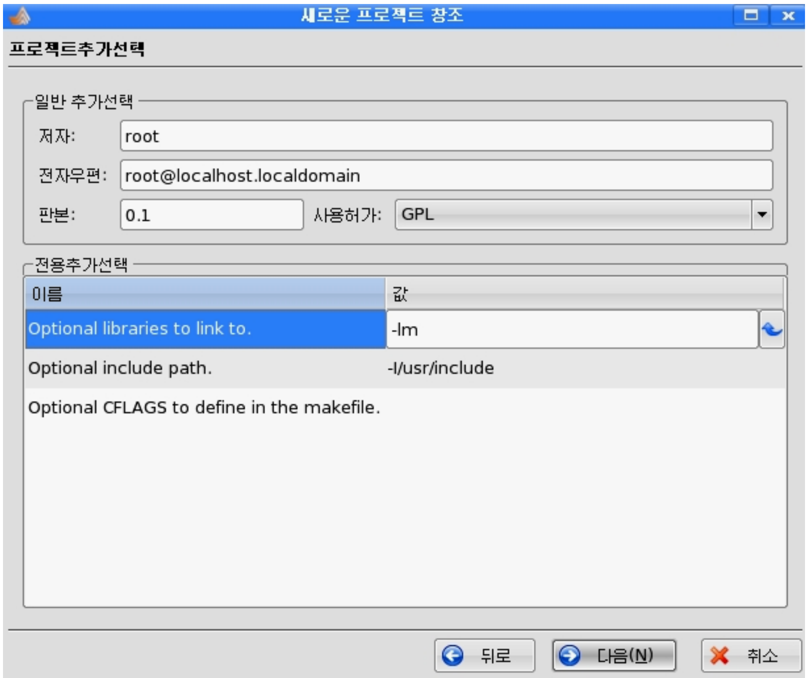
차림표띠에서 《프로젝트(P)》→《새 프로젝트(N)》지령을 실행하여《새로운 프로젝트 창조》대화칸을 펼친다.



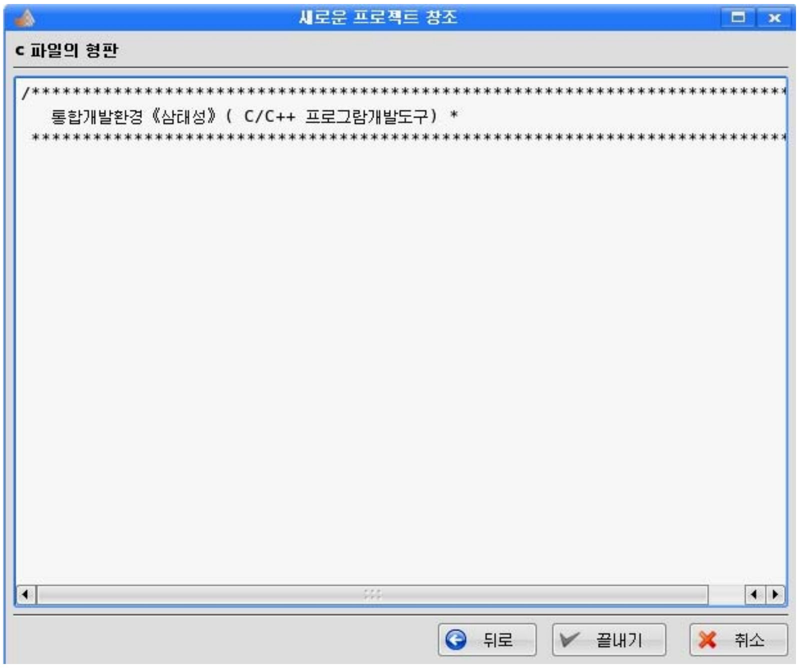


왼쪽의 《프로젝트형판들(V)》칸에서 C를 찰칵하여 나타나는 《간단한 Hello world프로그람》을 선택한다. 이때 오른쪽칸에는 프로그람의 실행결과와 그에 대한 해설문이 제시된다. 아래의 《속성》부분에서 《응용프로그람이름(N)》칸에 이제 작성할 프로그람의 이름 례로 hello를 입력하고 《위치(O)》칸에 프로그람을 보관할 등록부이름을 입력하면 《최종 위치》에 /root/hello와 같이 그 위치가 표시된다.

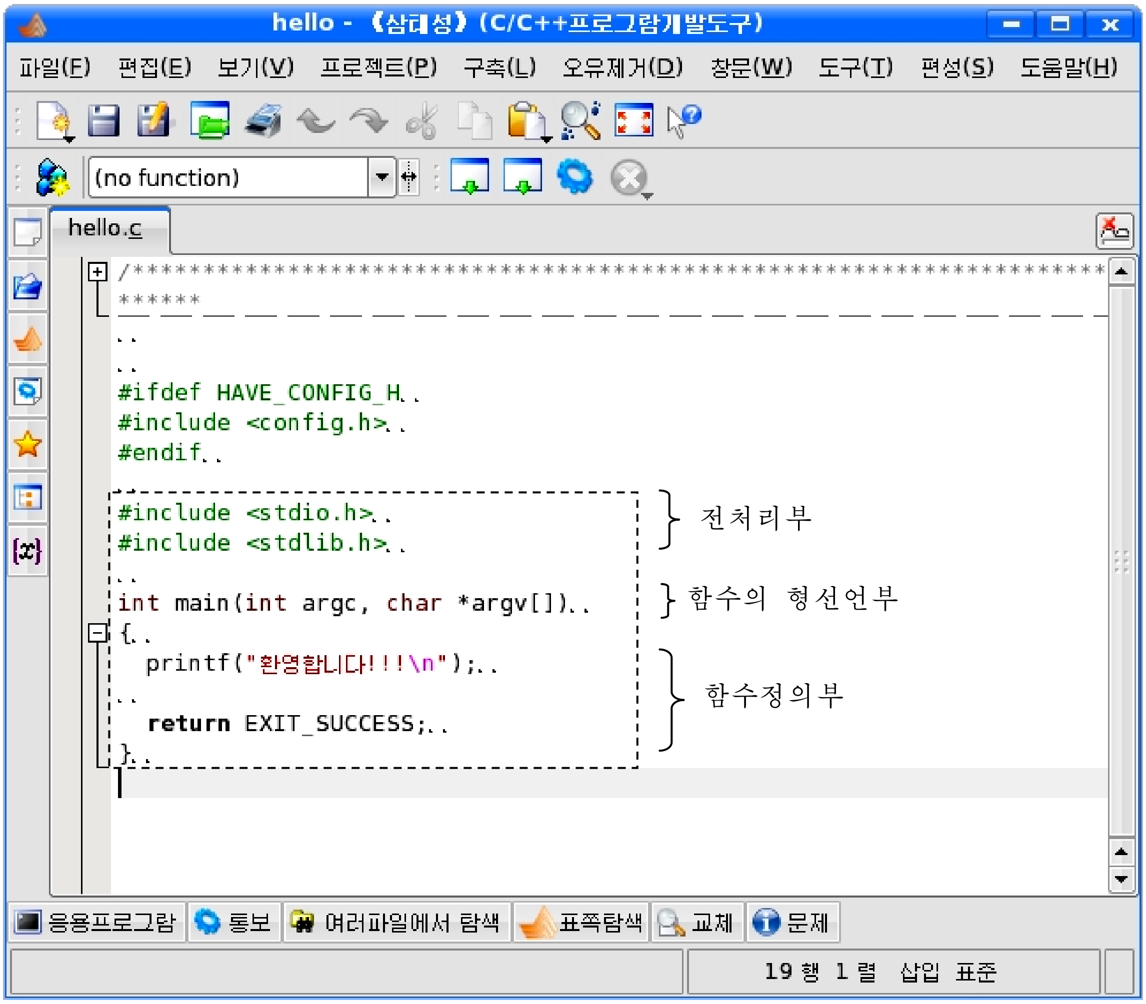
《다음(N)》단추를 찰칵하여 다음단계로 넘어간다.



《다음(N)》단추를 찰칵하여 다음단계로 넘어간다.



마지막으로 《끝내기》단추를 찰칵하면 개발도구창문에 다음과 같이 간단한 C프로그람이 제시된다.



그림에서 보는바와 같이 C프로그람은 전처리부, 함수의 형선언부, 함수정의부로 이루어지며 또한 변수정의부가 놓인다.

우의 례프로그람에서 #include <stdio.h>, #include <stdlib.h>와 같은 문장을 전처리문이라고 한다. 전처리문은 말그대로 프로그람의 번역전에 처리되는 명령문이다. 여기서 stdio.h, stdlib.h는 머리부파일의 이름이다.

머리부파일은 대체로 /usr/include보조등록부에 위치한다. 바로 이러한 머리부파일의 삽입이나 상수정의, 형정의를 진행하는것이 전처리부이다.

대역변수정의부에서는 대역변수가 정의된다. 대역변수는 프로그람안에 있는 모든 함수들에서 공동으로 리용되는 변수이다. 대역변수정의부는 전처리부다음에 함수의 형선언부앞에 놓인다.

례프로그람에서 int main(int argc, char *argv[])는 함수 main이 옹근수형이라는것을 선언한것이다. 이와 같이 함수의 형선언부에서는 함수들의 형태를 선언한다.

함수정의부에서는 함수의 본체를 정의한다.

C언어는 수속형언어로서 C프로그람은 어떤 독자적인 기능을 수행하는 명령문들의 묶음인 함수들로 이루어져있다.

여러개의 함수들가운데서 제일 먼저 실행되는 함수를 **주함수**라고 한다.

C프로그람은 적어도 한개의 함수를 가져야 하며 그것은 반드시 주함수이다.

주함수의 이름은 main으로 되여있으며 형식은 다음과 같다.

```
main([int argc, char *argv[]])
{
 명령블로크;
}
```

C언어로 작성된 함수들은 시작과 끝이 { }기호로 둘러막혀야 한다.

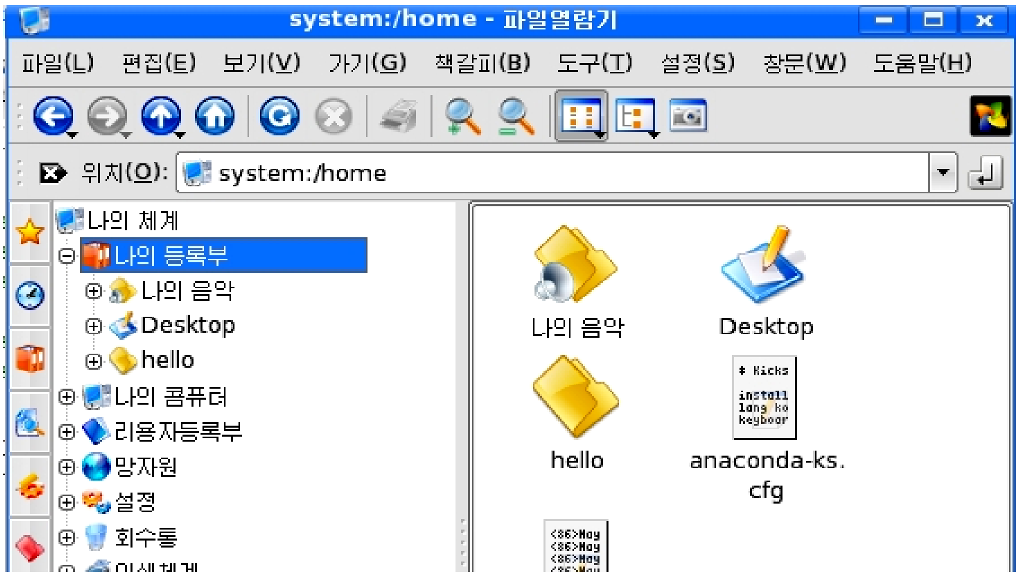
3. C프로그람의 개발과정

① 편집기를 리용하여 원천프로그람(source)을 작성한다.

원천프로그람은 프로그람언어를 리용하여 작성한 프로그람을 말한다. 여기에서는 C언어로 작성된 프로그람을 말한다.

우리 식 조작체계 《붉은별》에서는 앞에서와 같이 통합개발환경 《삼태성》의 C/C++프로그람개발도구창문을 리용하거나 《시작》→《응용프로그람》→《본문편집기》를 실행하여 기동되는 본문편집기에서 원천프로그람을 작성할수 있다.

앞에서 우리는 《간단한 Hello world프로그람》에 이름을 hello라고 달았는데 이 프로그람은 hello.c라는 파일이름으로 보관된다. C언어로 작성된 원천프로그람들은 확장자가 c로 된 파일이름으로 보관된다. 파일열람기에서 《나의 등록부》를 열어보면 hello라는 이름을 가진 등록부가 새로 생겨난것을 볼수 있다.



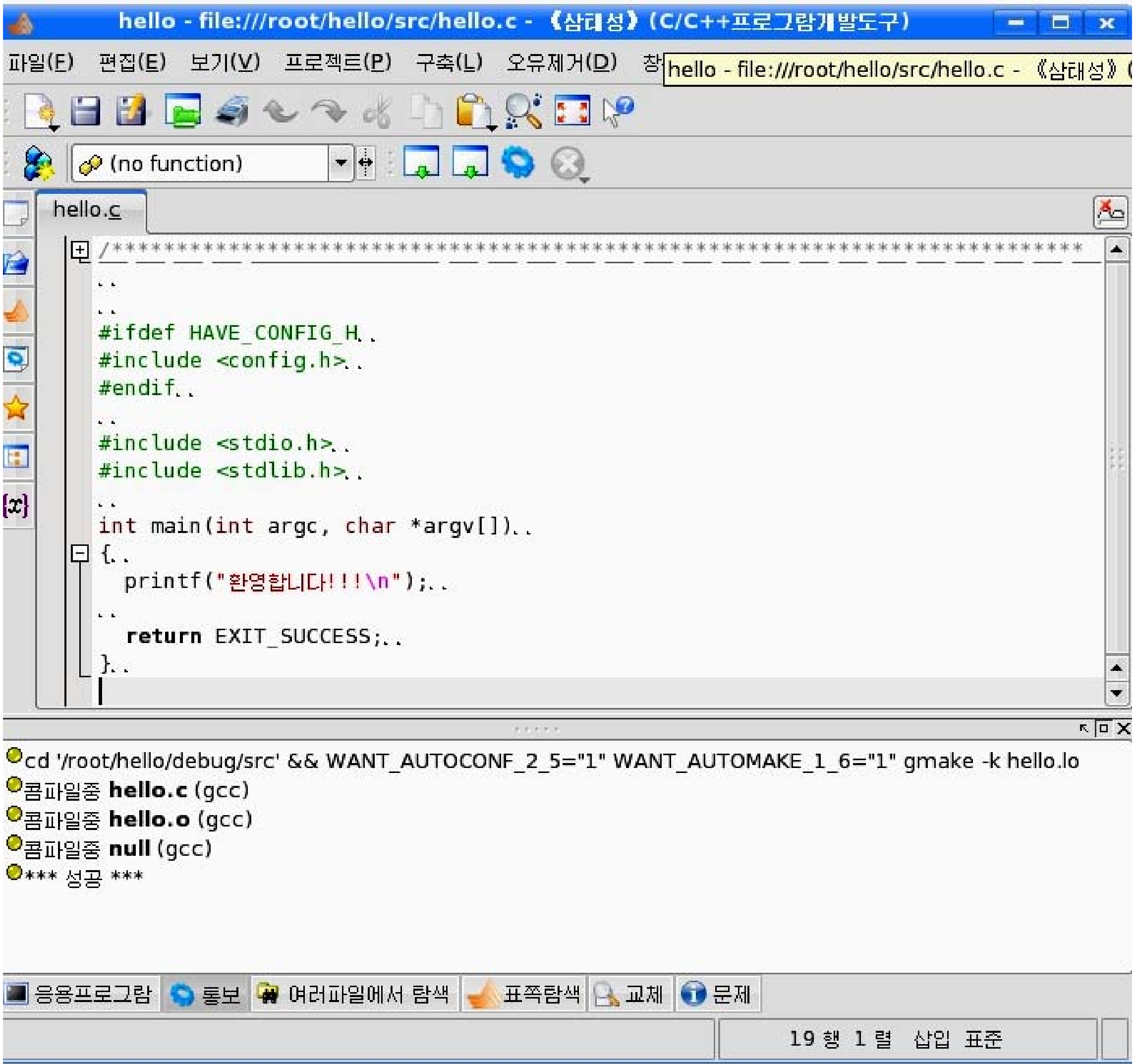
/hello/src등록부를 열어보면 hello.c파일이 있다.

② 원천프로그람의 번역 및 련결을 진행한다.

프로그람을 편집한 다음에는 그것을 번역 및 련결하여 실행가능한 파일로 만들어야 한다.

앞에서 작성한 《간단한 Hello world프로그람》의 번역 및 련결과정을 보기로 하자.

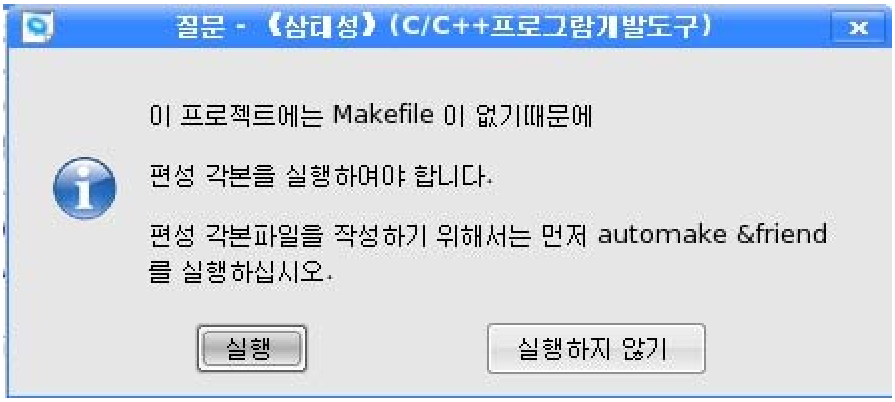
C/C++프로그람개발도구창문의 차림표띠에서 《구축(L)》→《파일 콤파일》지령을 실행하면 프로그람에 대한 번역을 진행하면서 창문의 아래쪽에 다음과 같이 통보가 제시된다.



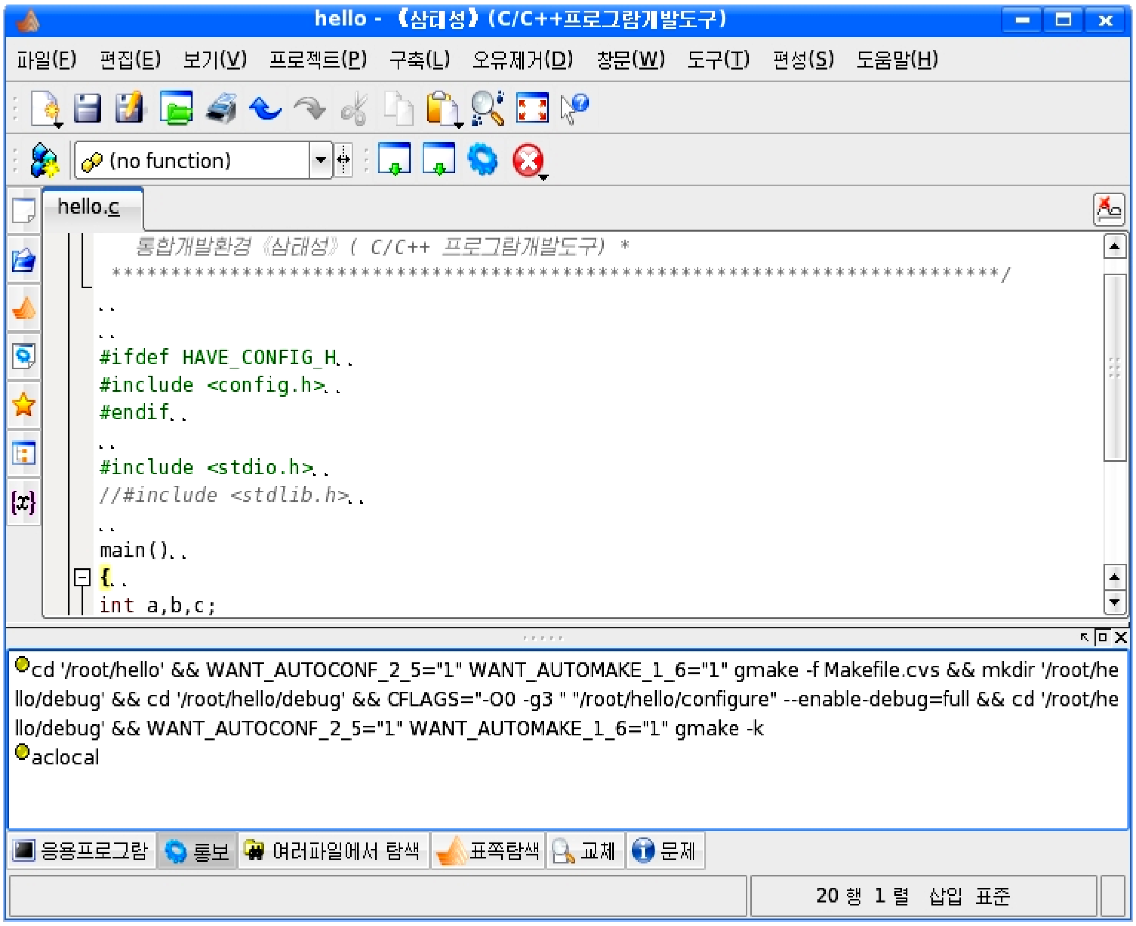
이 통보에서 보여주는바와 같이 C/C++프로그람개발도구창문에서 작성된 C프로그람은 Linux조작체계배포판들에서 제공되는 C를 위한 ANSI표준문법을 가지고있는 64bit번역기인 gcc(GNU C Compiler)를 리용하여 번역을 진행한다.

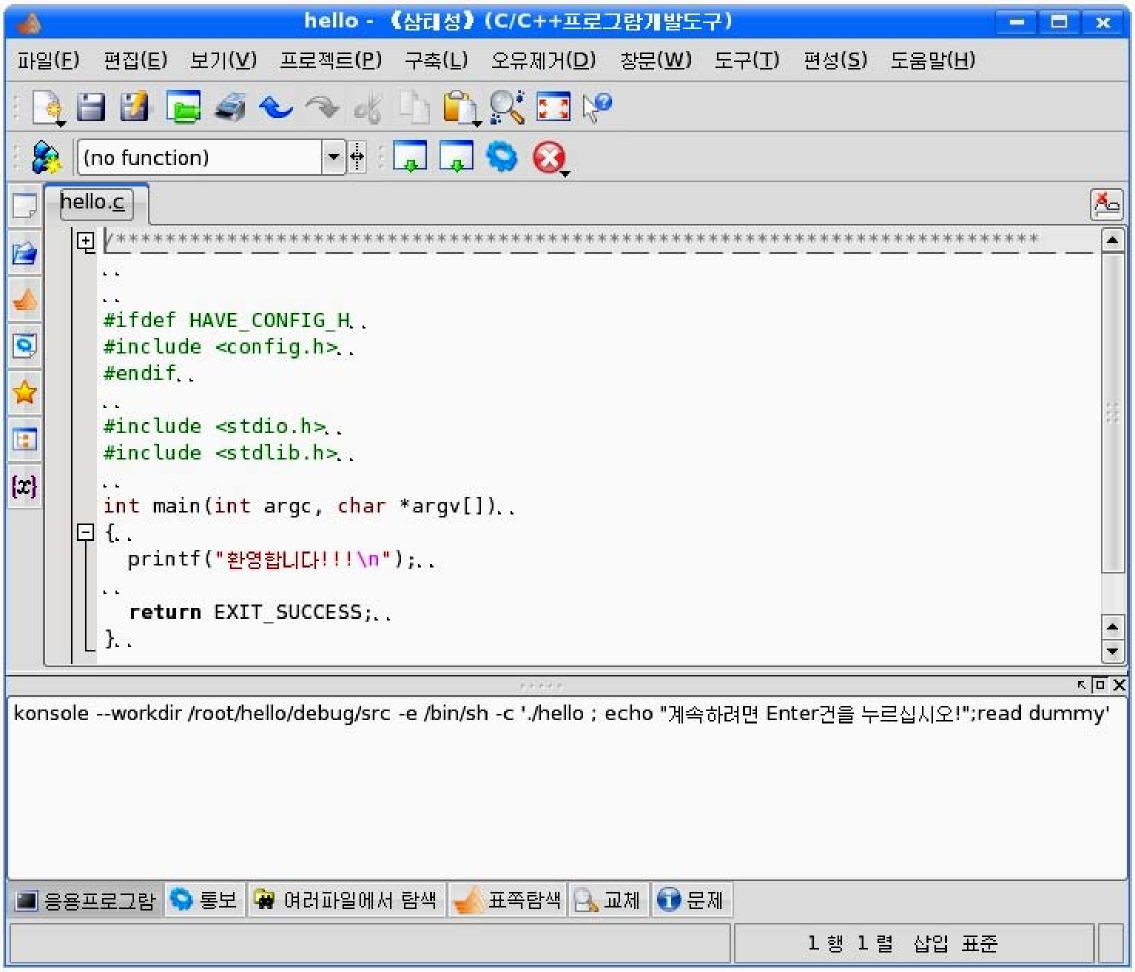
gcc는 대체로 /usr/bin이나 /usr/local/bin등록부에 기억되여있다.

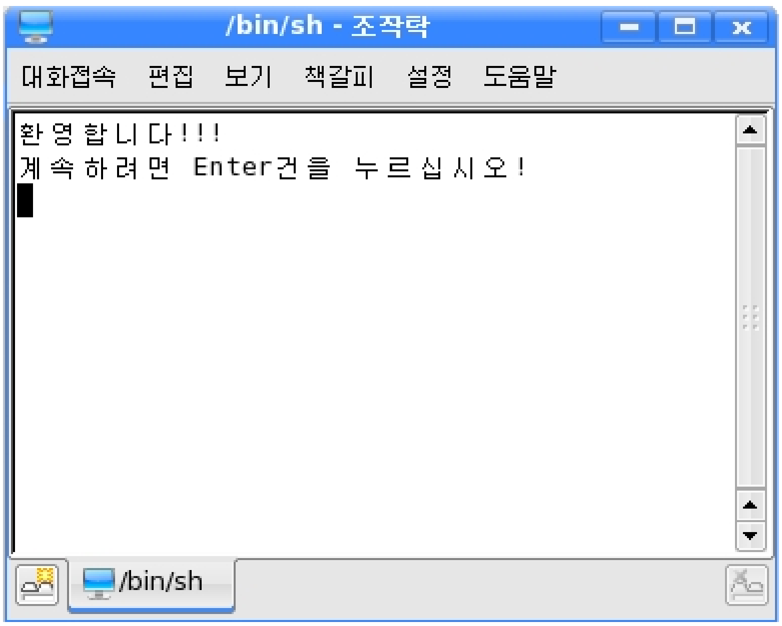
C/C++프로그람개발도구창문에서 처음으로 프로그람을 작성하고 차림표띠에서 《구축(L)》→《파일 실행》지령을 실행하거나 도구띠에서 프로그람실행단추 를 찰칵하면 다음과 같은 통보가 제시된다.



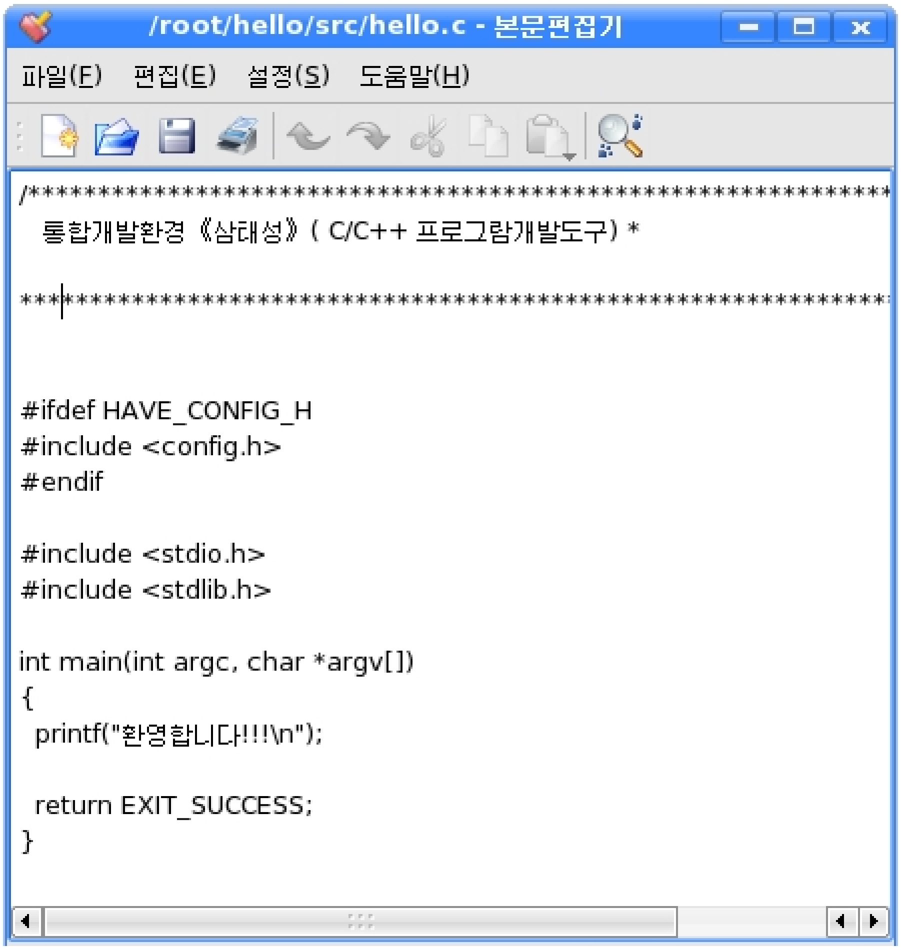
여기서 《실행》단추를 찰칵하면 우에서와 같이 프로그람에 대한 번역이 진행되고 번역이 성공하면 통보가 제시되며 조작탁창문에 결과가 현시된다.



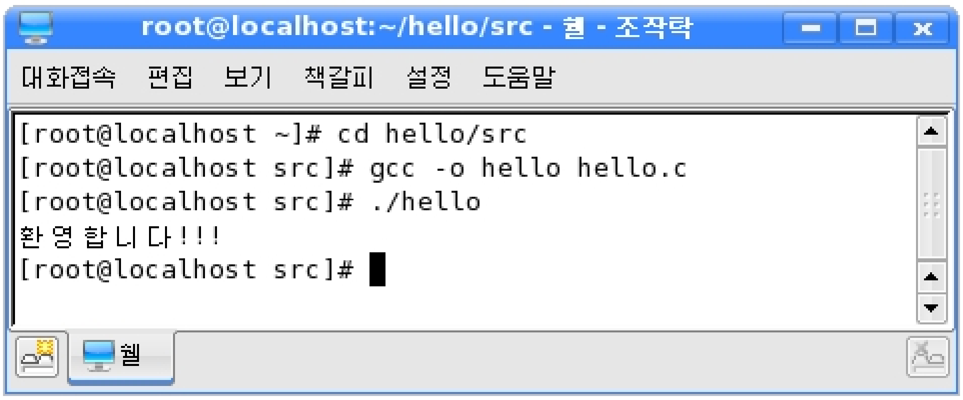

hello - 《삼태성》(C/C++프로그람개발도구)
파일(F) 편집(E) 보기(V) 프로젝트(P) 구축(L) 오유제거(D) 창문(W) 도구(T) 편성(S) 도움말(H)
(no function)
hello.c
#ifdef HAVE_CONFIG_H
#include <config.h>
#endif
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
printf("환영합니다!!!\n");
return EXIT_SUCCESS;
}
konsole --workdir /root/hello/debug/src -e /bin/sh -c './hello ; echo "계속하려면 Enter건을 누르십시오!";read dummy'
응용프로그람 통보 여러파일에서 탐색 표쪽탐색 교체 문제
1 행 1 렬 삽입 표준


/bin/sh - 조작탁
대화접속 편집 보기 책갈피 설정 도움말
환영합니다!!!
계속하려면 Enter건을 누르십시오!
/bin/sh

/hello/src등록부에서 hello.c파일을 두번찰칵하면 다음과 같이 본문편집기창문에 앞에서 작성한 《간단한 Hello world프로그람》이 입력되여있는것을 볼수 있다.



조작탁창문에서 cd hello/src지령을 입력하고 Enter건을 눌러 /src등록부를 작업등록부로 만든다. 다음 gcc -o hello hello.c지령을 입력하고 Enter건을 누르면 hello.c파일을 번역하여 hello라는 실행파일로 보관한다. 이때 hello라는 파일은 현재 작업등록부에 보관된다.



실행파일의 실행은 앞에 ./를 쓰고 그뒤에 실행파일의 이름을 쓰면 된다.

현재 작업등록부에서 ./hello지령을 실행하면 앞에서 만든《간단한 Hello world 프로그람》이 실행된다.

③ 오유수정도구를 리용하여 오유를 수정한다.

번역과정에 문법적오유가 검출되면 오유통보문이 현시되면서 처리가 중지되며 결과 실행파일이 생성되지 않는다. 이때에는 다시 편집기를 기동하여 오유통보문에 따르는 오유원인을 찾아 프로그람을 수정하여 다시 번역을 진행한다.

오유의 형태

· **문법적오유:** 언어에서 제정한 규칙에 어긋나게 표현된 부분으로서 이것은 번역시에 그 류형에 따라 행번호와 오유통보문으로 통지된다.

· **론리적오유:** 프로그람언어에서 제정한 규칙에는 어긋나지 않지만 프로그람적요구를 정확히 실현하지 못하게 하는 오유를 말한다.

이러한 론리적오유는 오유수정도구를 리용하여 발견하고 제거할수 있다.

Linux배포판이나 《붉은별》조작체계에서는 오유수정도구 gdb를 제공한다.

알아두기

gdb에서 많이 리용되는 지령

- list지령은 오유수정도구안에서 원천프로그람을 볼수 있게 한다.
- print지령은 변수검사지령으로서 프로그람에서 리용되는 변수들의 내용을 보기 위한 지령이다.
- run지령은 프로그람을 실행시키는 지령이다.
- break지령은 지정된 행이나 함수에 중단점을 설정하며 clear지령은 반대로 중단점을 해제하는 지령이다.
- info지령은 프로그람의 파라메터정보나 중단점, 출력설정들을 볼수 있게 하는 지령이다.
- set지령은 변수들의 값을 설정하는 기능을 수행한다.
- disassemble지령은 프로그람을 아쎔블리어로 현시한다.
- help지령은 gdb에 대한 도움말을 현시한다.





콤 퓨 터 상 식

통합개발환경 《삼태성》 설치법

1. 설치시 주의사항

《삼태성》2.0은 《붉은별》2.0(사용자용체계)에서 동작한다.

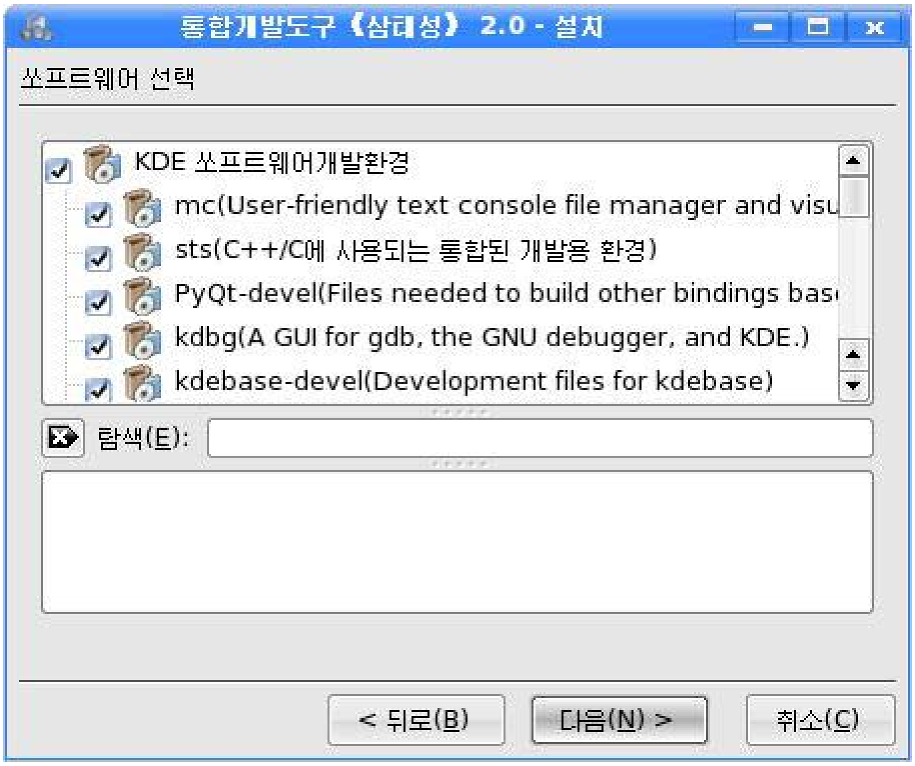
주기억 128MB이상, 동작주파수 1GHz이상, 하드디스크용량 20GB이상을 가진 Pentium Ⅲ이상의 콤퓨터에서 설치하여야 한다.

2. 설치

CD로 또는 하드디스크에서 설치할수 있다.

《삼태성》2.0이 들어있는 등록부에서 development.repo파일을 두번찰칵한다. 그러면 통합개발도구 《삼태성》2.0설치대화칸이 현시된다.

대화칸에서 《다음》단추를 찰칵하면 쏘프트웨어선택단계로 넘어간다.



쏘프트웨어묶음들을 선택한다. 개발도구의 충분한 기능을 리용하려면 모두 선택을 진행하여야 한다. 《다음》단추를 찰칵하면 설치될 쏘프트웨어묶음과 설치용량이 현시된다. 《다음》단추를 찰칵하면 실제적인 설치작업이 진행된다.

설치가 완료되면 성과적으로 설치되였다는 통보가 현시된다.

마지막으로 《완료》단추를 찰칵한다.

제2장. 간단한 프로그람작성

제1절. 두 수의 합계산

이 절에서는 두 수의 합을 구하는 프로그람작성과정을 통하여 C언어의 기본기호와 구성요소들을 학습하기로 한다.

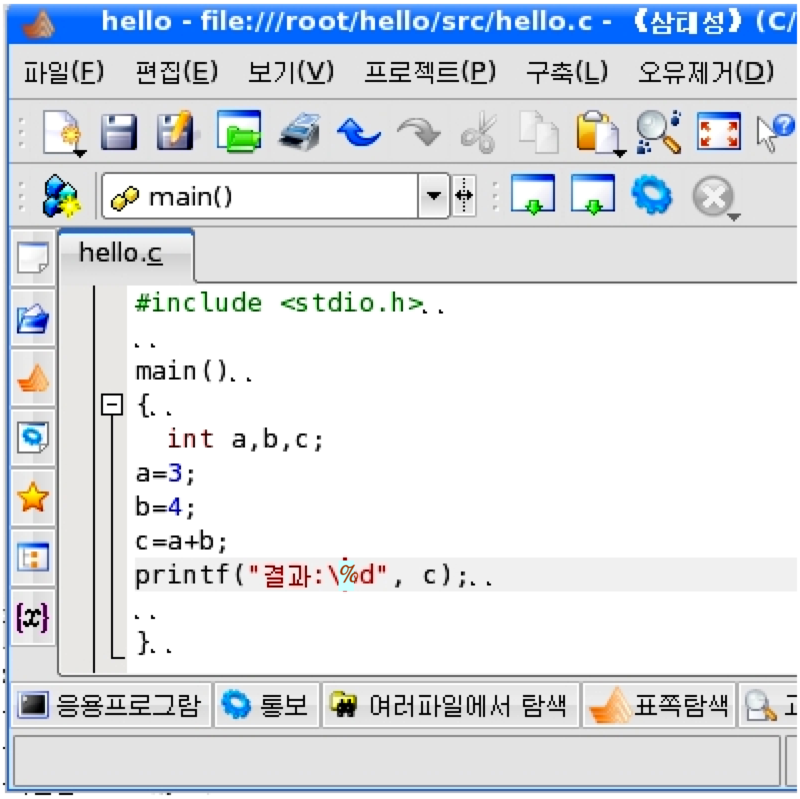
두 수 3과 4의 합을 계산하는 알고리듬을 만들자.

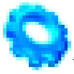
① 시작
② a←3, b←4
③ c←a+b
④ c 출력
⑤ 끝

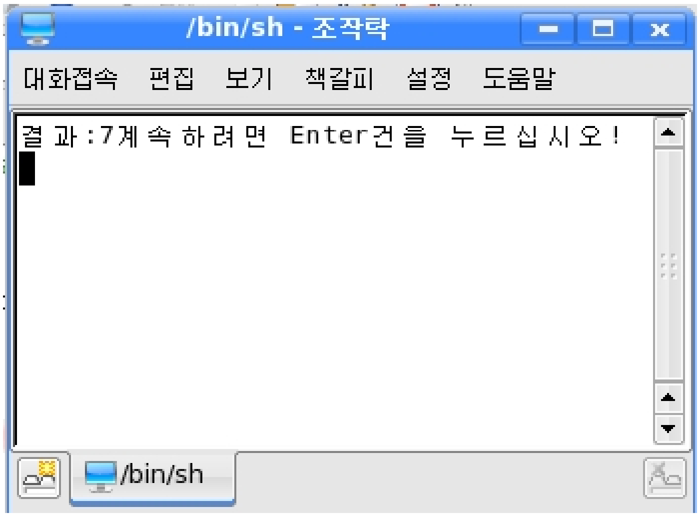
블로크도:

a ← 3, b ← 4
c ← a + b
c 출력

이 프로그람을 C/C++프로그람개발도구창문에서 작성하면 다음과 같다.



실행단추 를 찰칵하면 조작탁창문에 결과가 현시된다.



우의 프로그람에서 쓰인 모든 기호들은 C언어의 기본기호에 속한다.

－ C언어의 기본기호

① 영어문자: A, B, C, D, …, Z; a, b, c, d, …, z

② 수자: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

③ 기호: ! % ˆ & * ( ) + - = { } | ~ [ ] \ ; _
‘ “ : < > ? , . / #

－ 구성요소

우의 례에서 보는바와 같이 프로그람은 모두 여러개의 명령문들로 이루어져있으며 명령문은 또한 개개의 구성요소들로 이루어진다.

① 이름

이름은 상수나 변수, 함수, 표식 등 프로그람에서 사용되는 매개 요소들을 식별하기 위하여 주는 기호들의 렬이다. 따라서 프로그람안의 매개 요소들은 이름에 의하여 식별된다. 이름에는 표준이름과 사용자정의이름이 있다.

○ 표준이름

C언어처리기가 미리 정의하고 관리하는 이름으로서 여기에는 표준자료형이름, 표준상수이름, 표준변수이름, 표준파일이름 등이 있다. 표준이름은 미리 정의되여있기때문에 사용자들이 따로 정의하지 않고도 리용할수 있다.

○ 사용자정의이름

사용자정의이름은 사용자들이 정의해서 C언어처리기에 알려주고 리용하는 이름을 말한다. 함수나 변수 등의 이름은 프로그람작성자가 임의로 정의할수 있다.

○ C언어의 이름짓기규칙

- 이름은 영어문자나 또는 밑선으로 시작될수 있으며 그뒤에는 임의의 문자나 수자 또는 밑선을 놓을수 있다
- 대문자와 소문자는 각각 서로 다른 이름으로 해석된다. 례로 Index와 index는 각각 서로 다른 이름으로 된다.

· 이름은 수자로 시작하지 말아야 한다.

· 이름은 예약어와 같지 말아야 한다.

C언어에서 이름은 대체로 소문자를 많이 쓰며 대문자는 상수들과 일부 특수기호를 표현하는데 리용된다. 그러므로 프로그람을 작성할 때 될수록 이름을 소문자로 작성하여야 한다.

② 예약어

예약어란 미리 약속된 단어로서 C언어가 의미를 미리 정해놓은 단어를 말한다.

C언어에 정의되여있는 예약어

asm, auto, break, case, char, const continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

③ 설명문

설명문이란 프로그람실행과는 관련이 없는 문자렬로서 프로그람작성자들이 명령문이나 함수, 변수들에 설명을 달아주는것을 말한다.

설명문은 프로그람명령문과 구별하기 위하여 기호 /*로 시작하여 */로 끝난다. 번역프로그람은 이 부분을 콤퓨터가 아는 기계어로 번역하지 않는다.

두개 빗선 //도 행단위설명문으로 리용한다.

④ 분리기호

C언어에서 리용되고있는 분리기호들은 다음과 같다.

, { } ; = ( ) :





콤퓨터상식

네봐린나상

정보과학분야의 국제상은 없는가, 있다면 어디에서 취급하고있는가 하는 물음을 가지는 학생들도 있을것이다.

1980년대에 들어와 네봐린나상이라는것이 제정되였는데 이것은 정보과학의 수학적측면에 관한 업적을 대상으로 하여 1982년이래 국제수학자회의때마다 1명에게 수여되고있다. 수상자이름은 국제수학자회의 개회식에서 발표되고 수상식이 진행된다.

국제수학련맹은 유네스코의 자매기관인 국제학술련맹에 속하는 조직이며 수학분야의 국제협력을 추진하는 기관으로 되고있다. 국제수학련맹은 국제수학자회의를 4년마다 개최한다. 1988년도 국제수학회의는 베를린에서 열리였으며 2002년에 다음번 회의가 열렸다.

제2절. 두 수의 평균값계산

이 절에서는 두 수의 평균값을 계산하는 문제를 통하여 변수와 상수 그리고 자료형에 대하여 학습하도록 한다.

두 수 3과 4의 평균값을 계산하는 알고리듬을 만들자.

① 시작
② a←3, b←4
③ c←(a+b)/2
④ c 출력
⑤ 끝

블로크도:

a ← 3, b ← 4
c ← (a+b)/2
c 출력

프로그람의 실행전기간 변하지 않는 수인 즉 3과 4는 옹근수형상수이다.

1. 상 수

수학에서는 문제풀이과정에 항상 변화되지 않는 값을 상수라고 하는데 프로그람작성에서도 그 의미는 같다.

상수란 프로그람이 실행되는 전기간 변하지 않는 값(수나 문자렬)을 말한다.

C언어에서 모든 상수는 크기와 형을 가질뿐 문자렬상수를 제외하고는 기억기의 주소를 가지지 않는다. 즉 기억기에 기억되지 않는다.

상수는 크게 수값상수와 문자 및 문자렬상수로 나눈다.

1) 수값상수

수값상수는 옹근수형상수와 실수형상수로 분류된다.

① 옹근수형상수

옹근수형상수들은 보통 10진수로 표시한다. 그러나 필요에 따라서 2진수, 8진수, 16진수 등 임의의 진수로 표시할수 있다.

8진수는 첫머리에 영어소문자 o를 붙이며 0부터 7까지의 수자를 사용한다. 례를 들어 o127은 10진수 127이 아니라 8진수 127(=87_{10})이다.

16진수는 첫머리에 0x를 붙이며 0부터 9사이의 수자와 A, B, C, D, E, F의 영어문자를 사용한다. 례를 들어 0x100은 10진수 100이 아니라 16진수 100(=256_{10})이다.

long형의 옹근상수인 경우에는 수의 뒤끝에 L 또는 l표식을 덧붙인다.

례 1: o15L(8진수), 13L(10진수), 0xFFL(16진수)

② 실수형상수

실수형상수는 수학에서 소수점이 있는 수와 같다.
실수형상수에는 고정소수점수형식과 류동소수점수형식의 두가지가 있다.
실수형상수의 표시형식:

```
mmm.nn[ekkk]
   mmm.nn : 결수부
   kkk    : 지수부
   e      : 밑수 10
```

고정소수점수형식:

123.56, 3.141592

류동소수점수형식:

12.356e2(=12.356×10^{2}), 1235.6e-2(=1235.6×10^{-2})

2) 문자 및 문자렬상수

① 문자상수

문자상수는 ' '안에 넣는 문자이다.

례 2: 'a', '1', 'c'

문자상수는 실제적으로 아스키코드를 의미하므로 례에서 문자상수 'a'는 옹근수 96(0x61)과 같다. 마찬가지로 문자상수 '1'은 옹근수 48이므로 수자 1과 차이가 있다.

출력조종기호로 쓰이는 문자상수들은 빗선(\)으로부터 시작한다.

C언어에서 리용되는 특수한 문자상수(조종문자)들은 다음과 같다.

문자상수	기 능	아스키코드
\0	령문자로서 문자렬의 끝을 나타냄	0x00
\a	종소리(bell)	0x07
\b	뒤걸음치기(backspace건의 기능과 같다.)	0x08
\t	행은 그대로 있으면서 유표를 Tab만큼 이동한다.	0x09
\n	행바꾸기를 진행(유표의 렬위치는 변화되지 않는다.)	0x0A
\f	페지바꾸기를 진행	0x0C
\r	복귀한다.(Enter건의 기능과 같다.)	0x0D

② 문자렬상수

문자렬이란 하나이상의 문자들의 묶음을 말한다. 문자렬상수는 “ ”안에 포함된 기호렬이다.

례 3: “Korea is One!”
“K”
“This string is longer than that\n”
“124355676”

문자렬상수는 다른 형태의 상수들과는 달리 주기억기안에 기억되며 따라서 기억주소를 가진다.

문자렬상수는 주기억기에서 련속적인 바이트들에 차례로 기억되기때문에 문자렬상수의 첫 시작주소만 알면 해당 문자렬상수의 위치를 알수 있다.

문자렬상수에는 체계에 의하여 문자렬의 끝에 문자렬끝을 의미하는 령문자 \0이 자동적으로 삽입된다. 따라서 문자렬상수의 실제크기는 문자렬상수에 포함된 실제문자들의 개수보다 1개 더 큰 값으로 된다. 례로 “abc”라면 문자렬상수 abc의 실제크기는 4이다.

2. 변 수

앞의 알고리듬에서 a, b, c는 변수이다. 초기값을 다르게 주면 다른 값이 들어간다. 그러면 변수에 대하여 보기로 하자.

우리는 수학에서 변수라는 말을 많이 하여왔다. 수학에서의 변수는 변하는 수를 나타내는 문자를 말한다.

프로그람에서 변수의 개념은 수학에서의 개념과 비슷하다. 즉 프로그람에서 변화되는 값을 가질수 있는 문자 또는 문자렬을 말한다.

콤퓨터와의 관계속에서 고찰해볼 때 변화되는 자료를 보관하기 위한 장소는 주기억기이다.

기억기에 보관된 자료는 기억주소에 의하여 식별된다. 또한 프로그람에서 변화되는 값은 문자렬로 구성된 이름에 의하여 식별된다. 그러므로 프로그람에서의 변수는 한마디로 주기억기의 주소와 같다고 말할수 있다.

변수는 그 변수에 값주기하려는 자료의 형에 따라서 다시말하여 변수에 값주기될 자료의 크기에 맞게 기억공간을 차지할수 있다. 자료형은 뒤에서 학습한다.

프로그람에서 리용되는 변수의 선언은 자료형을 동반한다.

일반형식:

<자료형이름> <변수이름1>, <변수이름2>, … , <변수이름n>;

함수 및 변수의 이름은 프로그람작성자가 임의로 정의할수 있다.

- 이름은 영어문자나 또는 밑선으로 시작될수 있으며 그뒤에는 임의의 문자나 수자 또는 밑선을 놓을수 있다.
- 대문자와 소문자는 각각 서로 다른 이름으로 해석된다.

  C언어에서 이름은 대체로 소문자를 많이 쓰며 대문자는 상수들과 일부 특수기호를 표현하는데 리용된다. 그러므로 프로그람을 작성할 때 될수록 이름을 소문자로 작성하여야 한다.
- 이름은 수자로 시작하지 말아야 한다.
- C언어와 약속된 체계변수, 예약어, 표준함수들과 같은 이름으로 짓지 말아야 한다.

례:

Index
idex
SetVideoMode
_abc
_9Label
Label_9

} 옳다.

9_Label
*abc

} 옳지 않다.

변수의 분류는 상수의 분류와 같다.
C언어의 변수에는 구조체형변수가 있다.
이 부분에 대하여서는 뒤에서 보기로 한다.

3. 자 료 형

1) 자료형에 대한 일반개념

어떤 자료를 보관시키는데 얼마만한 기억용량이 필요한가를 결정하는것이 콤퓨터에서의 자료형에 대한 개념이다.

자료형이란 말그대로 자료의 형태로서 자료를 담을수 있는 그릇의 크기 다시말하여 프로그람실행시 자료를 기억시킬수 있는 기억기의 크기를 표현한것이다.

기억기에 할당된 크기에 의하여 자료를 보관할수 있는 값범위가 결정된다.

가령 1byte의 기억크기가 확보되였다면 $2^8 = 256$이므로 부호 없는 옹근수인 경우에는 0~255사이의 값을, 부호있는 옹근수인 경우 -128~127사이의 옹근수를 기억할수 있다. 이렇게 놓고볼 때 자료형은 자료를 기억시킬수 있는 값범위라고 말할수 있다.

프로그람을 작성할 때에는 여러가지 종류의 자료(례하면 수값이나 문자 등)들을 취급하게 된다.

례로 두 수의 합을 구하는 프로그람을 작성할 때 int a, b, c라고 하였는데 이것은 변수 a, b, c가 옹근수형변수라는것을 선언한것이다. 즉 이 변수들은 옹근수형의 값들만을 취급하며 이 변수들에 기억되여있는 값들을 옹근수로 해석하라는것이다. 그러면 그 자료형들을 구체적으로 보기로 하자.

2) 표준자료형

표준자료형이란 체계개발자들에 의하여 미리 정의된 자료형을 말한다.

C언어가 제공하는 표준자료형은 수값자료형들의 모임으로서 크게 소수점이 있는가, 없는가에 따라 옹근수형과 실수형으로 구분할수 있으며 부호가 있는가 없는가에 따라 부호있는 수값형과 부호 없는 수값형으로 나눈다.

부호 없는 수값형은 형이름앞에 예약어 unsigned를, 부호있는 수값형에는 예약어 signed를 붙인다. 생략하면 부호있는 수값형이다.

① 옹근수형

옹근수형은 소수점이 없는 자료를 기억할수 있는 형으로서 그것이 취할수 있는 수값범위에 따라 char형, short형, int형, long형으로 나눈다.

매 옹근수형에 따르는 수값범위와 기억기에서 차지하는 바이트수, 자료형의 이름은 다음과 같다.

형이름	부호 없는 수값범위	부호있는 수값범위	기억기크기 (byte)
char	0~255	-128~127	1
short	0~65 535	-32 768~32 767	2
int	번역기에 따라 다르다. 즉 16bit번역기이면 short형과 같고 32bit번역기이면 long형과 같다. gcc번역기는 64bit 번역기이므로 8byte크기를 확보한다.		
long	0~4 294 967 295	-2 147 483 648~2 147 483 647	4

례 1: 다음의 자료들은 어떤 형을 선택하여야 하겠는가?

a = 356 : unsigned short형
b = 34765 : unsigned short형
c = -128 : (signed)char형
d = 21475183 : long형

② 실수형

실수형은 소수점이 있는 수를 보관하는 자료형으로서 그것이 취할수 있는 값범위, 유효자리수에 따라 float형, double형, long double형으로 나눈다.

실수형에 따르는 수값범위와 유효자리수는 다음과 같다.

형이름	값범위	유효자리수	기억크기
float	3.4E38	7	4
double	1.7E308	15	8
long double	1.2E4 932	19	10

이상의 기본자료형들을 묶어보면 다음의 표와 같다.

자료형이름	변형된 이름	크기	표시범위
char	unsigned char	8	-128~127
int	signed signed int	번역기에 따라 다르다.	
short	short int signed short signed short int	16	-32 768~32 767
long	long int signed long signed long int	32	-2 147 483 648~ 2 147 483 647
unsigned char	-	8	0~255
unsigned	unsigned int	16	0~65 535
unsigned short	unsigned short int	16	0~65 535
unsigned long	unsigned long int	32	0~4 294 967 295
float		32	±3.4E38(7자리)
double		64	±1.7E308(16자리)
long double		80	±1.2E4 932(19자리)

(이것은 번역프로그람에 따라 달라질수 있다.)

③ 문자형

C프로그람에서 취급할수 있는 문자들로서는 언어의 기본기호에 해당한 문자들이 포함된다. 콤퓨터내부에서는 문자도 역시 0과 1의 비트렬로 표시된다. 현재 사용되고 있는 아스키코드체계에서는 한 문자당 8bit 즉 1byte를 할당하고있다.

례 2: 문자 a는 0x61
문자 A는 0x41

여기서 보여주는바와 같이 아스키코드체계에서 매개 문자는 1byte의 서로 다른 아스키코드값을 가지며 0x00 - 0xFF까지 최대 256가지의 문자를 표시할수 있다. 이로부터 문자형은 1byte의 크기를 가지며 형이름 char로 표시된다.

례 3:
```
char a, b, c;
a = 'X' ;      /* 0x58 */
b = 'Y' ;      /* 0x59 */
c = 'Z' ;      /* 0x5A */
```

여기서 변수 a, b, c는 형이름 char에 의하여 문자형변수로 되며 매 변수들은

1byte의 아스키코드값을 가질수 있다. 따라서 변수 a에는 문자 X의 아스키코드값 0x58이 기억되며 마찬가지로 변수 b와 c에도 각각 문자 Y와 문자 Z의 아스키코드값인 0x59, 0x5A가 기억된다.

문자형자료의 특징으로부터 문자형자료를 1byte의 옹근수형자료로 취급하는것도 가능하다. 례를 들어 프로그람

```
main(){
char c1, c2;
printf("%c%c\n", c1, c2);
printf("%d%d\n", c1, c2);
}
```

을 실행시키면 다음의 결과가 얻어진다.

```
 a b
97 98
```

아스키코드로 0~127개의 문자를 사용하면 바이트에서 제일 웃비트가 0이므로 %d를 리용하여 출력할 때 정의 옹근수가 출력된다. 만일 아스키코드로 128~255개의 문자를 사용하면 바이트의 제일 웃비트가 1이므로 %d로 출력할 때 부의 옹근수가 출력된다. 례를 들어 다음의 코드(프로그람토막)

```
char c = 130;
printf("%d", c);
```

을 실행시키면 결과는 -126이 얻어진다.

만일 부호처리를 진행하지 않으려면 char형의 변수를 unsigned char로 정의하여야 한다. 이때 char형변수의 값범위는 0～255이다.

signed char와 unsigned char의 의미와 사용법은 signed int, unsigned int와 같으며 다만 한개 바이트를 차지한다는데서만 차이난다.

3) 사용자정의형

체계가 정의한 표준자료형을 가지고서는 프로그람을 원만히 작성할수 없다.

례로 도서목록프로그람을 작성하려고 할 때 해당 책을 특징짓는 지표로서 책의 제목, 폐지수, 출판년도, 출판사, 집필자이름 등을 선정한다면 이러한 자료를 보관하기 위한 표준자료형은 C언어에 주어져있지 않다.

C언어는 체계가 정의한 표준자료형외에 사용자에 의해서 자료형을 창조할수 있도록 정의하는 방법을 제공하고있다.

사용자정의형이란 체계에 의해서가 아니라 사용자에 의하여 정의되는 자료형을 말한다.

사용자정의형에는 구조체형과 공용체형, 렬거형이 있다.

사용자정의형자료형에 대하여서는 뒤에서 구체적으로 보기로 한다.

4) 형변환

- 자동형변환

일반적으로 C언어는 자료의 형을 제한하지 않는 프로그람작성언어이므로 하나의 식에 여러가지 자료형을 가진 변수들을 섞어쓸수 있다. 즉 혼합연산이 가능해진다.

C언어는 혼합연산이 진행될 때 자동적으로 높은 자료형으로 변환해준다.

자료형은 기억기의 크기를 결정하므로 작은 크기로 할당된 기억기의 내용을 그보다 큰 기억공간에 보관할수 있지만 자기보다 작은 기억공간에는 보관할수 없다. 왜냐하면 자료를 잃어버리기때문이다. 일상생활에서 10L짜리 바께쯔에 5L의 물을 담을수 있지만 5L짜리 바께쯔에 10L의 물을 담을수 없는 원리와 같다. 그러므로 낮은 자료형에서 높은 자료형으로의 변환은 자동적으로 진행된다.

례 4:
```
#include <stdio.h>
main()
{
    int a, b;
    double  c, d;
    unsigned char e;
    a = 500;
    b = 7;
    e = 100;
    c = a + e;
    d = 2 * c;
    printf("%d%d%f%f\n", a, b, c, d);
}
```

이 프로그람을 실행하면 500, 7, 600.0, 1200.0이라는 값이 출력된다.

명령문 c = a + e; 에서 변수 c는 double형, a는 int형, e는 char형이므로 자동적으로 더 높은 자료형인 double형의 연산결과로 현시된다.

- 강제형변환

다음의 코드를 고찰하자.

```
double  y;
y = 1/3;
```

이 명령문을 실행한 결과 y에는 0이 값주기된다. 그것은 연산이 옹근수 1과 옹근수 3의 나누기이므로 옹근수상인 0이 y에 값주기되기때문이다.

그러나 y = 1.0/3; 으로 쓰면 y에는 0.33이라는 값이 값주기된다. 왜냐하면 1.0이라는 실수의 나누기를 진행하였기때문이다.

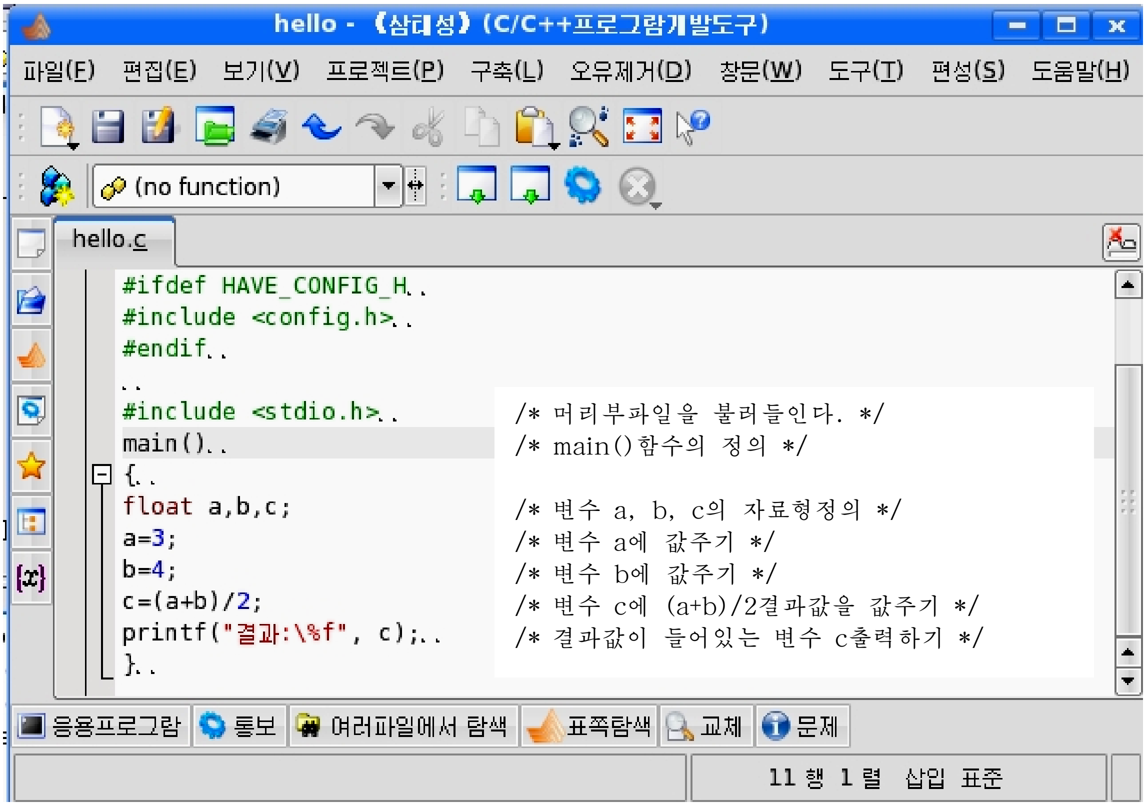
이와 같은 현상을 없애기 위하여 C언어는 강제형변환기능을 제공한다.

자료앞에 변환하려는 형예약어를 붙여서 요구하는 형으로 변환하는것을 **강제형변환**이라고 한다.

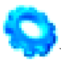
례 5: y = (float)1/3;

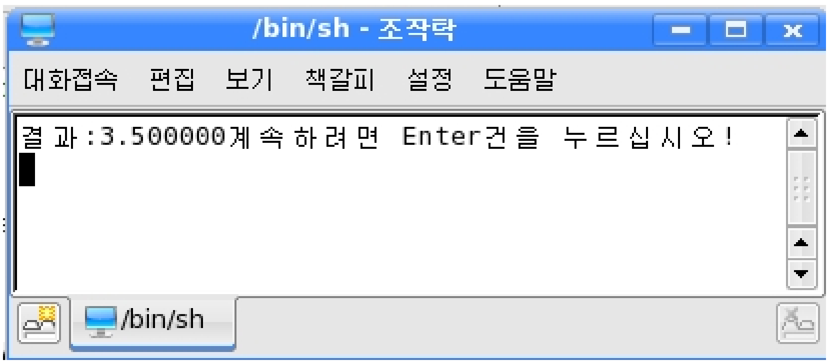
이외에도 형변환함수들을 리용하여 형변환을 진행할수 있다.

앞에서 배운 지식에 기초하여 두 수 3과 4의 평균값을 계산하는 프로그람을 다음과 같이 작성할수 있다.



이 프로그람에서 float a,b,c; 명령문은 변수 a, b, c의 자료형이 실수형인 float 형이라는것을 선언한것이다.

실행단추 를 찰칵하면 조작탁창문에 결과가 다음과 같이 현시된다.



련습문제

1. 다음의 변수들의 이름이 옳은가 틀리는가? 그 근거를 밝히여라.

ㄱ) _1234　　ㄴ) &xyz　　ㄷ) abc
ㄹ) struct　　ㅁ) 18cb　　ㅂ) %y

2. 다음의 수들을 기억하려면 몇byte이면 충분하겠는가?

ㄱ) 103　　ㄴ) -129　　ㄷ) 37 855
ㄹ) -65 500　　ㅁ) -32 760　　ㅂ) +255

3. 다음의 자료들을 관리하기 위한 자료형을 선택하여라.

ㄱ) 15　　ㄴ) -358　　ㄷ) 8.57　　ㄹ) 6.5×10^{-8}
ㅁ) 756 398　　ㅂ) +135 687　　ㅅ) “korea”　　ㅇ) “123”





콤퓨터상식

사람의 뇌수와 콤퓨터의 기억능력

콤퓨터가 사람의 뇌수와 같은 능력을 가지려면 어느 정도의 기억용량이 필요하겠는가?

연구자료에 의하면 콤퓨터에서 1초동안에 실행되는 연산회수와 기억용량이 수자상으로 거의 일치하군 하였는데 이 비률은 콤퓨터를 만가동시킬 목적에서 설정된것들이였다.

만약 이와 같은 비률을 념두에 둔다면 1초동안에 10조회의 연산을 진행하는것으로 보는 사람의 뇌수와 맞먹는 규모의 콤퓨터에서는 10조단어의 기억장치 다시말하여 약 1 000조bit정도의 기억용량이 필요하게 된다. 그것은 콤퓨터에서는 한 단어가 16bit로부터 64bit사이에 놓이는것이 보통이기때문이다.

연구자들이 연구한데 의하면 현재 첨단수준의 과학연구사업들에 사용되고있는 콤퓨터의 능력이 대체로 곤충의 신경계통정도의 수준이라는것을 알수 있었다고 한다.

콤퓨터의 끊임없는 수준향상으로 콤퓨터능력이 20년마다 1 000배의 속도로 증가하고있으며 이 속도로 나간다면 인간의 뇌수와 대등한 수준의 콤퓨터에 요구되는 초당 10조연산이라는 처리속도는 2030년경까지는 개인용콤퓨터에서도 가능해지는것으로 된다.

제3절. 두 수의 크기비교

이 절에서는 두 수의 크기를 비교하는 례를 통하여 C언어에서 연산자의 개념과 분류 그리고 산수, 비교, 값주기연산자에 대하여 학습하기로 한다.

두 수 a, b의 크기를 비교하여 그 결과를 출력하는 프로그람을 작성하자. 알고리듬은 다음과 같이 작성할수 있다.

① 시작
② a에 값주기
③ b에 값주기
④ a와 b의 크기비교
⑤ 출력
⑥ 끝

C언어에는 두 수의 합과 차의 계산, 크기비교 등을 할수 있는 여러가지 연산자들이 있다.

1. 연산자의 개념과 분류

상수, 변수, 함수와 연산자를 결합하여 표현한 값을 **식**이라고 한다.

콤퓨터로 문제를 푼다고 할 때 그것은 각이한 형태의 식을 계산하여 결과를 얻어나가는 과정이다. 이와 같이 어떤 식을 계산하여 결과를 얻어내는 과정을 **연산**이라고 한다.

연산에는 산수연산, 문자렬연산, 비교연산, 론리연산, 비트연산, 주소연산 등이 있다.

연산에 참가하는 기호를 **연산자**라고 하며 연산의 대상을 **연산수**라고 한다.

연산수가 1개인가 2개인가에 따라 단항연산자, 2항연산자, 3항연산자라고 한다.

례: A = B + C

여기서 A, B, C는 연산수이고 =와 +는 연산자이다.

C언어에는 연산의 종류에 따라 수많은 연산자들이 있다. 이것이 C언어의 중요한 우점이다.

C언어가 제공하는 연산자들을 합리적으로 리용하면 프로그람이 간소화되며 따라서 프로그람의 번역속도나 실행속도가 훨씬 빨라진다.

2. 산수연산자

산수연산은 산수식을 계산하여 수값형태의 자료를 얻는 연산이다.

산수식은 두개 혹은 그이상의 연산수들을 산수연산자와 괄호로 련결하여놓은것을 말한다.

산수연산을 표현하는 연산자를 **산수연산자**라고 한다.

C언어가 제공하는 산수연산자에는 수학에서와 같이 더하기(+), 덜기(-), 곱하기(*), 나누기(/)를 진행하는 4칙연산자외에 나머지연산을 진행하는 나머지연산자(%)가 있다.

+	더하기	A + B	A와 B의 합
-	덜 기	A - B	A와 B의 차
*	곱하기	A * B	A와 B의 적
/	나누기	A / B	A를 B로 나눈 상
%	나머지	A % B	A를 B로 나눈 나머지

연산을 진행할 때 괄호 ( )를 리용하여 우선권을 보장할수 있다.

산수연산자들은 모두 2개의 연산수를 가지는 2항연산자들이다.

례 1: y * (x + 3)
a + b
-x + y
y mod 6

어떤 변수에 하나를 더하거나 더는 특별한 단항연산자로서 증가연산자(++)와 감소연산자(--)가 있다.

일반형식:

++ : 증가연산자

-- : 감소연산자

```
례 2: a++;    /* a=a+1 */
      a--;    /* a=a-1 */
```

증가, 감소연산자는 연산수의 앞에 놓일수도 있고 연산수의 뒤에 놓일수도 있다. 이 경우 식평가에서 차이가 있다.

례 3: ++a: a=a+1을 진행한 다음 그 값을 식에 적용한다.
a++: a의 값을 식에 적용한 다음 a=a+1을 진행한다.

```
례 4: int a = 1, b, c;
      b = a++;        /* b=1 */
      c = ++a;        /* c=3 */
      printf("b=%d\n",b);
      printf("c=%d\n",c);
```

3. 비교연산자

우리는 수학에서 $x + y < 1.5$라든가 $x \geqq -10.5$ 와 같은 같기식, 안같기식들에 대하여 배운다.

비교연산은 같기식, 안같기식에 대응하는 개념으로서 자료의 크기를 비교하는 연산이다. 비교연산을 표현하는 연산자를 **비교연산자**라고 한다.

C언어가 제공하는 비교연산자는 다음과 같다.

의　미	수　학	C언어	실　례
작기	<	<	A<B
작거나 같기	≦	<=	A<=B
크기	>	>	A>B
같거나 크기	≧	>=	A>=B
같기	=	==	A==B
같지 않기	≠	!=	A!=B

비교연산자들은 모두 2항연산자들이다.

비교연산에서는 다음과 같은것을 주의하여야 한다.

① 비교연산수들은 반드시 같은 형으로 되여야 한다.

② 비교연산을 할 때 수값은 수의 크기에 따라, 문자렬은 문자의 아스키코드값에 따라 크기를 결정한다.

③ 비교연산의 결과는 1과 0만을 가진다. 즉 참(true)인 경우에는 1, 거짓(false)인 경우에는 0을 준다.

례 1:
```
5 < 3              ⇒ 0 (false)
"bad" > "back"     ⇒ 1 (true)
a > b              ⇒ a가 b보다 큰가?
x == y + 1         ⇒ x가 y+1과 같은가?
a * b > c          ⇒ a*b가 c보다 큰가?
```

례 2: 비교연산자를 리용하는 프로그람

```
#include <stdio.h>
void main(void)
{
    int a=1,b=2;
    printf("%d<%d ? %d\n", a,b,a<b);
    printf("%d<=%d ? %d\n", a,b,a<=b);
    printf("%d>%d ? %d\n", a,b,a>b);
    printf("%d>=%d ? %d\n", a,b,a>=b);
}
```

```
1<2 ? 1
1<=2 ? 1
1>2 ? 0
1>=2 ? 0
```

4. 값주기연산자

값주기연산자는 어떤 변수에 값을 넣어주는 연산자이다.

기본값주기연산자는 =이다.

=는 오른변의 값을 왼변에 넣기하는 기본값주기연산자이다.

일반형식:

V = e;
V: 변수이름, e: 식, = : 값주기연산자

례 1: a=3; /* 변수 a에 3을 값주기한다. */

c=x*y; /* 변수 c에 x와 y를 곱한 값을 값주기한다. */

값주기연산자는 산수연산자(+, -, *, /), 비트연산자(&, |, ^, <<, >>)와 조합하여 식을 간단하게 하면서도 여러가지 연산을 진행하게 한다.

여기서는 산수연산자와의 조합만을 보기로 한다.

의미	기호	식
곱하기 값주기	*=	a*=b: a=a*b
나누기 값주기	/=	a/=b: a=a/b
나머지 값주기	%=	a%=b: a=a%b
더하기 값주기	+=	a+=b: a=a+b
덜기 값주기	-=	a-=b: a=a-b

일반적으로 다른 프로그람언어들에서는 값주기연산을 x=x+a로 쓰지만 C언어에서는 간단히 x+=a로 쓸수 있다.

이와 같이 두개의 연산자를 결합하여 사용함으로써 프로그람코드의 표현을 줄일수 있다.

례 2: A*=(B=C+D); 는 다음과 같은 연산을 묶은것과 같다.

```
{ B=C+D;
{ A=A*B;
```

이 례에서 본것처럼 하나의 명령문에 여러 값주기연산자가 있어도 되지만 그 연산순서를 ( )로 정확하게 표현해주어야 한다.

련습문제

1. 다음 식에서 연산순서를 밝히고 C언어에서 제공하는 연산자로 표시하여라.

   1) $0.2a - \frac{2}{5}b + \frac{1}{2} + b\frac{4}{5} + a$

   2) $3a - \{2b + [4a - (-b - 3a) + b + (a + 2b - 3c)]\}$

   3) $\frac{x^2 - y^2}{2}$ 4) $1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$

   5) $\frac{x^2}{a^2} + \frac{x^2}{b^2} + \frac{x^2}{c^2}$ 6) $(\frac{xy}{yb})\ z - b$

2. 다음 식에서 연산순서를 밝히고 연산결과를 쓰시오.

   1) 10 + 100 / 2 % 4 * 3 - 6 / 3 * 2

   2) 30 - (5 * 5 * 5) % 10 / 5 + 7 / 2





콤퓨터상식

소년시절 가우스의 한가지 일화

프로그람에서 절차가 긴 렬로 서술되는 처리는 경우에 따라 간단한 조작으로 같은 결과가 나오도록 수학적으로 변환할수 있다.

유명한 수학자 가우스는 소년시절에 학교에서 뛰여난 학생이였다고 한다.

한번은 교원이 재미절반으로 학생들을 향하여 1부터 100까지의 옹근수를 전부 더하라는 문제를 냈다. 학생들은 수자를 하나하나 더해나가는데 여념이 없었으므로 긴 계산시간이 필요되였지만 가우스만은 순식간에 정확한 해답을 냈다. 그는 100까지의 수자를 50개의 쌍 즉 1+100, 2+99, 3+98, …로 가르고 모든 쌍이 101이라는것에 주목하였다. 101을 50배 하면 5050이며 이 방법이면 그 많은 더하기계산을 하지 않고도 해답을 구할수 있었다. 이러한 계산의 고속화의 수법은 복잡한 콤퓨터처리에서도 중요하게 나선다. 이른바 《최적화콤파일러》는 고속화를 위한 변환용목록을 가지고있으며 경우에 따라 매우 근본적인 변화에 의하여 번역도중에 프로그람을 능률적으로 만든다. 여기서 열쇠로 되는것은 가우스가 한것처럼 계산의 순서와 자료의 표현을 전면적으로 재구성하는것이다. 우에서 언급한 소년시절 가우스에 대한 일화는 후에 알려진데 의하면 사실이 외곡된것이라고 한다. 실제로 교원이 제시한 문제는 우에서 언급한것보다 훨씬 더 힘든 문제였는바 81297+81495+81693+ … +100899를 구하는것이였다고 한다. 이 문제를 소년시절 가우스가 우의 방법으로 순식간에 암산하였다는것이다.

제4절. 삼각비계산

이 절에서는 삼각비를 계산하는 문제를 통하여 전처리문과 입출력함수 그리고 수값처리표준함수들에 대하여 보기로 한다.

각 x의 값을 입력하면 그에 대한 시누스, 코시누스, 탕겐스값을 구하는 프로그람을 작성하자. 알고리듬은 다음과 같이 작성할수 있다.

① 시작
② 각 x의 값을 입력
③ a←sin(x)
④ b←cos(x)
⑤ c←tan(x)
⑥ a, b, c출력
⑦ 끝

C언어에는 삼각함수값을 계산하는 함수가 표준적으로 제공되여있다.

1. 전처리문

1) 전처리문에 대한 개념

앞의 례들에서는 항상 프로그람의 본체를 작성하기 앞서 #include <stdio.h>라는 명령문을 써넣었다. 이것은 stdio.h라는 파일을 삽입하라는 뜻이다.

전처리문(preprocessor)이란 번역프로그람이 프로그람을 번역하기에 앞서 진행하여야 할 처리내용을 지시하는 명령문이다.

전처리문을 리용하면 지정된 파일의 삽입이나 문자렬교체(치환) 등의 조작들을 쉽게 할수 있다. 이것은 C언어에서 제공하는 가장 편리한 기능의 하나이다.

C언어의 명령문들은 반두점(;)으로 구분하면서 한 행에 여러개의 명령문들을 쓸수 있다.

그러나 전처리문은 한 행에 한개의 명령문만을 놓을수 있다. 만일 전처리문이 한 행을 벗어날 때에는 행의 끝에 빗선 \을 붙이고 다음행에 계속 쓰면 된다.

전처리문에는 파일삽입문, 문자렬치환문, 조건부번역문 등이 있다.

여기에서는 파일삽입문과 문자렬치환문만을 보기로 한다.

2) 파일삽입을 위한 전처리문

파일의 삽입은 전처리문예약어 include(포함하다)를 리용한다.

일반형식:

```
#include <파일이름>
#include "파일이름"
```

<파일이름>으로 지정하면 미리 번역프로그람에게 지시되여있는 등록부로부터 파일을 검색한다. 번역프로그람에는 서고파일이나 머리부파일의 등록부위치(경로)가 등

록되여있다. 그러므로 C언어에서 제공하는 표준함수를 리용할 때에는 <파일이름>형식으로 파일의 삽입을 진행한다.

"파일이름"형식의 파일삽입은 번역프로그람에 등록되여있지 않는 등록부로부터 파일의 검색을 진행하는 경우이다.

례 1:
```
#include <stdio.h>
#include <math.h>
#include "d:\usr\include\to.h"
```

파일의 삽입을 위한 전처리문을 프로그람에 써놓으면 번역프로그람은 번역하기 전에 먼저 그 위치에 해당한 파일을 삽입하고 그다음에야 원천파일과 함께 번역한다.

```
#include <stdio.h>
main()
{
  char file[64];
  … … …
  if (NULL==(fp(…)))
  … … …
}
```

stdio.h 에 있는 내용이 삽입 →

```
#define NULL 0
#define EOF (-1)
    … …
main()
{
char file[64];
… … …
if (NULL==(fp(…)))
… … …

}
```

알아두기

파일의 삽입은 왜 진행하는가?

C언어가 제공하는 모든 표준함수들은 확장자가 lib인 서고파일에 들어있다. 이 서고파일에 있는 함수들의 선언은 머리부파일 *.h에 있으므로 프로그람작성시 C언어가 제공하는 표준함수를 리용하려면 함수선언부가 있는 머리부파일을 삽입하여야 한다.

파일의 삽입은 또한 프로그람의 공동리용을 위하여 필요하다. 어떤 기능을 수행하는 프로그람이 존재할 때 그것을 다시 작성하지 않고 반복리용한다면 프로그람개발속도를 높일수 있다. 이와 같은것을 프로그람의 공동리용이라고 말한다.

례로 두 학생이 n!과 관련한 문제를 푼다고 할 때 두 학생이 각기 n!을 작성할 필요가 없이 한 학생이 작성한것을 서고파일로 만들어놓고 공동으로 리용한다면 프로그람의 리용률을 높일수 있을것이다.

3）문자렬치환을 위한 전처리문

문자렬치환은 어떤 수값이나 문자렬과 같은 상수들이나 식을 알기 쉬운 문자렬로 바꾸어 프로그람을 쉽게 해석할수 있게 하며 프로그람에서 여러번 쓰이는 상수들을 정의하는데서 매우 편리하다.

문자렬치환은 전처리문예약어 define(정의하다)을 리용한다.

일반형식:

```
#define 식별자  상수
#define 식별자  식
```

문자렬치환전처리문이 있으면 번역프로그람은 프로그람본체에서 나오는 모든 식별자들을 상수나 식으로 바꾸어준다.

```
#define SIZE  10
main()
{
  char file[SIZE];
  … … …
  for (i=0; i<SIZE; i++)
  … … …
}
```

⇒

```
main()
{
char file[10];
… … …
for (i=0; i<10; i++)
… … …

}
```

례 2:
```
#define  LENGTH  100
#define  MAX(A, B)  (((A) >= (B))?(A):(B))
#define  VOL(R)   (R*R*R)
```

문자렬치환시 파라메터를 가지는 문자렬은 ()로서 막는다.(파라메터에 대한 개념은 뒤에서 본다.)

#undef전처리문은 #define문으로 지정한 치환을 중지하게 한다. 즉 #undef전처리문이 나타난 시점에서부터는 치환하지 않는다.

례 3:
```
#define TRUE  -1      /* TRUE를 –1로 치환 */
#define FALSE TRUE  /* FALSE를 TRUE로 치환한 결과 -1로 인식된다.*/
#define  BOOL  int       /* BOOL형을 int형으로 치환 */
  int a;
     … … …
#undef  BOOL
#define  BOOL   char   /* BOOL을 char로 다시 치환 */
```

례 4: 0℃부터 100℃까지 10℃마다 셀씨우스온도(℃)로부터 화렌하이트온도(°F)로 변환하는 온도변환표를 작성하는 프로그람은 다음과 같다.

```
#include <stdio.h>
#define  LOWER  0
```

```
#define  UPPER  100
#define  STEP  10
main()
{
    int c;
    for (c=LOWER; c<=UPPER; c+=STEP)
    printf("%d%d\n", c, c*9/5+32);
}
```

2. 표준함수

표준함수는 C언어프로그람개발자들에 의하여 미리 정의된 전용의(특정한) 기능을 수행하는 함수로서 C언어프로그람개발환경에 들어있는것이 아니라 서고파일로 준비되여있다. 표준함수는 서고에 준비되여있으므로 필요한 표준함수를 리용하려면 전처리문을 리용하여 서고파일의 위치를 알려주어야 한다.

1) 입출력표준함수

(1) 형식화된 입출력함수

- 형식화된 자료출력함수(printf: print formatted)

일반형식:

```
printf("출력형식", 식1, 식2, …, 식n)
```

기능: 출력형식지정에 의하여 문자나 문자렬, 수값을 비롯한 계산결과를 출력한다. 출력형식은 다음과 같다.

```
%[-] [<마당의 크기>] <자료형식> [\조종기호]
```

자료형식에는 다음의 기호가 리용된다.

%d	부호있는 10진옹근수로 출력
%u	부호 없는 10진옹근수로 출력
%o	8진수로 출력
%x	16진수로 출력
%f	고정소수점형식의 출력
%e	류동소수점형식의 출력
%g	%f와 %e가운데서 보다 짧은쪽으로 출력
%c	문자로 출력
%s	문자렬로 출력

조종기호는 출력조종을 진행할뿐 화면에는 표시되지 않는다.
조종기호는 다음과 같다.

\n	행바꾸기	\"	"를 표시
\t	수평으로 8문자 뛰여넘기	\'	'를 표시
\f	페지바꾸기	\\	\를 표시
\r	되돌아가기	\ddd	3자리 8진수표시(아스키코드)
\b	한문자 앞으로 이동	\xhh	16진수표시(아스키코드)

마당크기: 오른쪽, 왼쪽을 기준으로 자료를 출력하기 위해서는 마당크기를 지정할수 있다. 자료의 출력을 진행할 때 출력형식을 %d, %e, %f를 지정하면 유효길이만큼 출력하기때문에 출력되는 값에 따라 표시되는 위치가 변한다. 이것을 피하기 위하여 마당크기를 지정한다.

%의 오른쪽에 -를 붙이면 그에 대응하는 값은 마당의 왼쪽끝에 맞추어 출력된다. 생략하면 오른쪽을 기준으로 표시된다. 마당의 크기보다 유효길이가 짧으면 그만큼 공백이 놓인다.

소수점이 있는 경우에는 표시한 마당의 전체길이와 소수점아래에 표시될 자리수를 지정할수 있다. 즉 마당의 크기를 다음과 같이 지정한다.

<마당전체의 길이>.<소수점아래의 길이>

표시할 자료가 문자렬이면 문자렬의 일부만을 표시할수 있다.

<마당전체의 길이>.<표시할 문자수>

결과 지정된 문자렬의 왼쪽에서부터 지정된 문자개수만큼 표시된다.
printf함수는 머리부파일 stdio.h에 선언되여있다.
이 함수를 리용하자면 다음과 같은 명령문을 프로그람의 선두에 삽입하여야 한다.

```
#include <stdio.h>
```

례 1: 다음의 프로그람은 화면에 어떤 형식으로 자료를 출력하는가?

```
#include <stdio.h>
void main(void)
{
  printf("          11111111112222222223\n");
  printf("123456789012345678901234567890\n");
  printf("- - - - - - - - - - - - - - - - - - - - -\n");
  printf("%-10d %10d\n",12345,12345);
  printf("%10s %-10s\n","ABCDEFGHI", "abcdefghi");
  printf("%-15e %10e\n",123.456,0.123e-3);
  printf("%-15.5s %10.5s\n",
           "ABCDEFGHIJKLMNO", "abcdefghijkl");
}
```

결과:

```
          11111111112222222222223
        123456789012345678901234567890
        - - - - - - - - - - - - - - - - - - - -
        12345               12345
        ABCDEFGHI abcdefghi
        1.23450e+002    1.230000e-004
        ABCDE               abcde
```

례 2: 화면에 C programming Language라고 표시하는 프로그람은 다음과 같다.

```
#include <stdio.h>
main()
{
   printf("C programming Language\n");
}
```

례 3: 출력결과가 다음과 같은 프로그람을 작성하면 아래와 같다.

```
ABCD
   EFG
              HIJKL     MNO
                         PQR
STUV          WXY
```

```
#include <stdio.h>
main()
{
   printf("ABCD\n");
   printf("   EFG\n");
   printf("              HIJKL     MNO\n");
   printf("                         PQR\n");
   printf("STUV          WXY\n");
}
```

- 형식화된 자료입력함수

일반형식:

scanf("입력형식", 변수1, 변수2, …, 변수n)

기능: 입력형식에 기초하여 건반으로부터 자료를 입력하여 지정된 변수들에 보관한다. 입력형식은 printf함수와 같다.

지정된 변수에로 건반자료가 입력되므로 그 변수가 차지하고있는 주소가 요구된다. 그러므로 변수목록에 들어가는 매 변수(지적자변수가 아닐 때에만)이름의 앞에 주소연산자 &를 붙여주어 주소를 지정한다.

scanf함수에서는 인쇄조종기호 \n을 사용하지 않는다.

scanf함수도 머리부파일 stdio.h에 선언되여있다.

알아두기

scanf함수를 사용하는데서 주의할 점

① scanf()함수가 형식에 기초한 입력으로 생각하고 입력형식화안에 문자렬을 쓸수 없다. 가령 입력하기 전에 문자렬통보문을 현시할것을 생각하고 printf 함수처럼 scanf("Input number:%d" , &n); 라고 쓰면 오유가 발생한다. 문자렬출력은 printf()함수를 리용하여야 한다.

② scanf()함수가 실행되여 자료를 입력해야 할 때 아무것도 입력하지 않고 Enter건을 누르면 변수에 아무것도 들어가지 않는다.

③ scanf()함수에 의하여 하나의 변수에 수값을 입력할 때 수자들사이에 공백이 있으면 공백앞까지의 자료만 변수에 들어간다. 례로 13 45라고 입력하면 변수에는 13만 입력된다.

례 4: 건반에 의하여 3자리의 10진수를 읽어서 그것을 변수 x에 넣고 다시 현시하는 프로그람

```
1:  #include <stdio.h>
2:  main()
3:   {
4:       int x;
5:       scanf("%3d", &x);
6:       printf("%3d\n", x);
7:   }
```

해 설

1행에 서술된 명령문은 프로그람에서 리용되는 표준입출력함수들의 선언을 위한 머리부파일삽입을 진행한다.

4행에서는 문제의 요구에 따라 변수 x를 선언한다.

5행에서는 10진수 3자리를 읽기 위하여 입력형식지정에서 마당크기를 3d로 지정하였다.

6행에서는 역시 3자리로 출력형식을 진행하여 출력한다.

례 5: 두 수 a와 b를 입력하여 그의 적, 상, 합, 차, 나머지를 구하고 화면에 출력하는 프로그람

```
#include <stdio.h>
main()
{
     int a,b;
     scanf(“%d%d”, &a, &b);
     printf(“%d×%d=%d\n”, a,b,a*b);
     printf(“%d÷%d=%d\n”, a,b,a/b);
     printf(“%d+%d=%d\n”, a,b,a+b);
     printf(“%d−%d=%d\n”, a,b,a-b);
     printf(“%d≡%d=%d\n”, a,b,a%b);
}
```

례 6: 여러가지 자료형을 리용한 례

```
#include <stdio.h>
main()
{
     long a;
     unsigned  int b;
     char c;
     a=-120000;
     b=0x3FFF;
     c=o15;
     printf(“value a is %d\n”, a);
     printf(“value b is %x\n”, b);
     printf(“value c is %o\n”, c);
}
```

해 설

b=0x3FFF; 명령문에서 0x는 뒤의 수자가 16진수임을 나타내며 c=o15; 에서 o는 뒤의 수자가 8진수임을 나타낸다.

(2) 문자, 문자렬입출력함수

- 한 문자 입력

한 문자의 입력에는 %c의 지정으로 scanf()를 사용할수도 있지만 보다는 getchar()함수를 많이 사용한다.

례 7:
```
char c;
scanf("%c=", &c);
printf("c=%c", c);
c=getchar();
printf("c=%c", c);
```
getchar()함수는 눌리운 건에 해당한 아스키코드값을 돌려준다.

- 한 문자 출력

getchar()함수의 경우와 마찬가지로 한 문자 출력도 %c를 지정한 printf()함수보다 putchar()함수를 많이 사용한다.

례 8:
```
char c;
c=getchar();
printf("%c", c);
putchar(c);
```
- 문자렬의 입력과 출력

문자렬의 입력과 출력에는 gets()와 puts()를 사용한다.

례 9: 문자렬을 입력하고 그것을 다시 출력하는 프로그람

```
#include <stdio.h>
void main(void)
{
char buf[50];
gets(buf);
puts(buf);
}
```

```
I am a pupil.
I am a pupil.
```

례에서 보는바와 같이 gets()는 공백을 포함한 한개의 행을 그대로 파라메터로 넘겨진 buf에 보관한다.

2) 수값처리표준함수

절대값함수 abs()를 제외한 수값처리표준함수들은 머리부파일 math.h에 선언되여있다.

C언어가 제공하는 수값처리표준함수들은 다음과 같다.

함수	X의 형	Y의 형	의 미	비 고
abs(x)	int		절대값계산	
exp(x)	double		e^x계산	
log(x)	double		$\log_e x$	x > 0
log10(x)	double		$\log_{10} x$	x > 0
pow(x, y)	double	double int	x^y	x > 0, (x, y)≠0
sqrt(x)	double		$\sqrt{x}$	x ≧ 0
fabs(x)	double		실수 x의 절대값	
floor(x)	double		x를 넘지 않는 가장 큰 옹근수	
ceil(x)	double		x보다 작지 않은 옹근수	
sin(x)	double		sinx	
cos(x)	double		cosx	
tan(x)	double		tanx	
asin(x)	double		arcsinx	
acos(x)	double		arccosx	
atan(x)	double		arctanx	

례 10: 다음의 경우 y값은 어떤 값을 가지는가?

y=abs(-3) : y에 3이 값주기된다.
y=exp(2.5) : y에 $e^{2.5}$이 값주기된다.
y=log(10.8) : y에 $\log_e 10.8$이 값주기된다.
y=log10(10.8) : y에 $\log_{10} 10.8$이 값주기된다.
y=pow(5, 8.5) : y에 $5^{8.5}$이 값주기된다.
y=sqrt(100.5) : y에 $\sqrt{100.5}$가 값주기된다.
y=fabs(-8.8) : y에 8.8이 값주기된다.
y=floor(3.85) : y에 3이 값주기된다.
y=ceil(3.85) : y에 4가 값주기된다.

례 11: 다음의 프로그람에서 틀린것을 지적하여라.

```
#include <math.h>
```

```
#include <stdio.h>
main()
{
   double a=3.14, b=-3.14, c=2.5, d, e, f, g;
   int i=2;
   d=pow(a,b);
   e=pow(b,c);
   g=abs(b);
 }
```

오유는 다음과 같다.

8행에서 b가 부수이므로 pow()함수를 리용할수 없다.

9행에서 b가 실수이므로 abs()함수를 리용할수 없다. 즉 fabs()함수를 리용하여야 한다.

각 x의 값을 입력하면 그에 대한 시누스, 코시누스, 탕겐스값을 구하는 프로그람은 다음과 같다.

```
#include <stdio.h>
#include <math.h>
main()
{
float x,a,b,c;
scanf("%f", &x);
a=sin(x);
b=cos(x);
c=tan(x);
printf("sin(x)=%f\n", a);
printf("cos(x)=%f\n", a);
printf("tan(x)=%f\n", a);
}
```

련습문제

1. 다음 프로그람의 실행결과를 밝히여라.

ㄱ)
```
#include <stdio.h>
main()
{
  int a, b;
  a=1500;
  b=350;
  printf("%d/%d=%d……%d\n", a,b,a/b,a%b);
}
```

ㄴ)
```
#include <stdio.h>
main()
{
int a, b, c, d;
a=d=3, b=1, c=6;
printf("(%d-%d)×(%d-%d)/(%d+%d)=%d\n", a,b,c,d,a,d,
                                              (a-b)*(c-d)/(a+d));
}
```

ㄷ)
```
#include <stdio.h>
main()
{
float a=3.5, b=2.9, c=5.1;
float m;
m=(a+b+c)/(c-b-a);
printf("(%f+%f+%f)÷(%f-%f-%f)=%f\n", a,b,c,c,b,a,m);
}
```

ㄹ)
```
#include <stdio.h>
main()
{
 int  i, j, x, y, z;
 i=7;
 j=5;
 x=i>j;
 y=i<j;
 z=i==j;
printf("i>j= ⇒ %d\n", x);
printf("i<j= ⇒ %d\n", y);
printf("i=j= ⇒ %d\n", z);
}
```

ㅁ)

```
#include <stdio.h>
main()
{
  int a=5, b=9;
  printf("%d,%d\n", ++a,a);
  printf("%d,%d\n", a++,a);
  printf("%d,%d\n", --b,a);
  printf("%d,%d\n", b--,a);
}
```

2. 다음의 기능을 수행하는 간단한 프로그람을 작성하여라.

1) X = -4, Y = 6, Z = 7일 때 (X * Y)/(X + Z)를 구하여라.
2) 15 + 313 - 214 + (95 / 5) = 의 뒤에 결과를 표시하여라.
3) 9800/8800의 상과 나머지를 구하여라.
4) 329분은 몇시간 몇분인가를 구하는 프로그람을 작성하여라.
5) 한 변의 길이가 a인 바른4각형의 면적을 출력하는 프로그람을 작성하여라.
6) 9 801을 화면의 15번자리에 표시하는 프로그람을 작성하여라. 오른쪽에 표시할 때 왼쪽 11자리를 0으로 채우도록 한다.
7) 3.141 592를 왼쪽을 기준으로 하여 소수점아래 4자리까지 표시하여라.
8) 16진수 3A1E를 10진수로 화면에 표시하여라.
9) 8진수 7206을 16진수로 화면에 표시하여라.
10) 10진수 8 801을 8진수와 16진수로 화면에 표시하여라.
11) 16진수연산 7AF - 3BC를 구하여 16진수, 10진수로 화면에 표시하여라.
12) 30 × 20의 결과를 16진수 3자리로 표시하여라.
13) 3각형의 세 변의 길이가 주어졌을 때 면적을 구하여라.
14) 직3각형의 두 직각변이 주어졌을 때 빗변의 길이를 구하여라.
15) 반경 r가 주어졌을 때 원의 면적과 둘레의 길이를 구하여라.
16) 반경 r와 높이 h가 주어졌을 때 원기둥의 겉면적과 체적을 구하여라.
17) 반경 r가 주어졌을 때 구의 겉면적과 체적을 구하여라.
18) 두 밑변과 높이가 주어질 때 제형의 면적을 구하여라.
19) 반경 r인 원에 내접하는 바른4각형의 면적을 구하여라.
20) 한 변의 길이가 a인 바른3각형에 내접하는 원의 면적과 둘레의 길이를 구하여라.
21) 웃반경, 아래반경, 높이가 주어졌을 때 원뿔대의 체적을 구하여라.
22) 직각자리표계에서 두 점의 자리표가 주어졌을 때 두 점사이의 거리를 구하여라.
23) 직각자리표계에서 세 점의 자리표가 주어졌을 때 3각형의 면적을 구하여라.
24) 세 수 a, b, c가 주어졌을 때 합평균, 곱평균을 구하여라.
25) 반경이 r1, r2, r3인 세 원이 서로 접하고있을 때 세 원의 중심점을 정점으로 하는 3각형의 면적을 구하여라.

3. 다음 식을 C언어의 식으로 표시하여라.

1) $(y \div x)^{n-1}$
2) $(x^2 - y^2) \times (1 \div 2)$
3) $x^2 / a^2 + y^2 / b^2 + z^2 / c^2$
4) $1 / a^2 \times (b / 2) \times n - 1$
5) $a - 1 + b / (1 + x^2) + |x - 3| / 2$
6) $(2 / px) \times \sin x$
7) $((e^x - e - x) / 2) + ((e^x + e - x) / 2)^2$
8) $(x - y)\sin x + (x^2 + y^2)\tan x$
9) $|\ln x^2 - y| + 7.3x - 3y = 2x + 3$
10) $|a| \times |b|\cos A$
11) $2x + b / (1 + x^2) + |x - a| / 2$
12) $xy / x^2 + y^2 - \ln(|x - (1 + x^2) / 2|)$
13) $1/2 \lg((1 + \sin x) / (1 - \sin x))$
14) $e - 1 / x (x - m)^2 + 5x\sin x^3$
15) $(\ln(x - y) / 2 - \sin z) - 2$
16) $\lg(x^2 + 1) / x^2 + 2\arctan(1 - a) / 2$





콤퓨터상식

《진실》과 《거짓》의 수학

현대의 수자형콤퓨터에서 연산은 2진수 즉 수자 1과 0의 두 값에 의하여 진행된다. 이런 연산에 기초한 수자형콤퓨터는 20세기 중엽에 만들어졌지만 이것을 취급할수 있는 수학적방법은 이미 1800년대 중엽에 영국의 수학자이며 론리학자인 죠지 불에 의하여 확립되였다. 이 론리연산법을 《불대수》라고 한다.

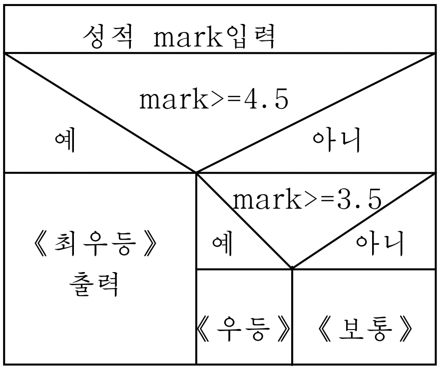
《불대수》에서는 변수와 명제가 《진실》인가 《거짓》인가의 어느 한 값만을 가지며 연산을 론리적, 론리합, 론리부정 등의 연산자에 의하여 론리적으로 표시한다. 1과 0으로 값을 표시한다는것은 전기회로스위치의 《투입》과 《차단》에 대응시킬수 있기때문에 론리적이나 론리합 그리고 론리부정 등의 론리연산을 그대로 전자회로로 실현할수 있다. 즉 론리연산에 기초한 계산이나 판단을 장치에 의하여 자동화할수 있는셈이다. 그리하여 《불대수》는 콤퓨터에 쉽게 응용될수 있었다. 그러나 150여년전에 이 연산법을 내놓은 불자신은 생전에 오늘과 같은 수자형콤퓨터의 출현에 의하여 자기의 연구성과가 쓰이리라고는 예견할수 없었을것이다.

제5절. 학생성적평가

이 절에서는 학생성적평가프로그람을 작성하는 과정을 통하여 론리연산자와 조건명령문들에 대하여 학습하기로 한다.

학생의 성적을 입력하여 성적을 《최우등》, 《우등》, 《보통》으로 평가하는 프로그람을 작성하여라. 알고리듬은 다음과 같다.

<알고리듬>



알고리듬에서 보는바와 같이 성적을 입력하여 먼저 성적이 4.5이상인가 아닌가를 판단하고 성적이 4.5보다 작으면 다시 3.5이상인가 아닌가를 판단하여 결과를 출력하여야 하므로 여기에서는 조건이 주어졌을 때 둘중의 하나의 처리를 진행하는 명령문이 필요하다. 조건식의 판단은《옳다》또는《틀린다》의 두 결과를 가지며 두 결과에 대하여 어느 한 처리를 진행하는데 쓰이는 명령문이 바로 조건명령문이다. 이 절에서는 론리연산과 조건판단을 진행하여 여러개가운데서 한 처리를 진행하는 명령문에 대하여 학습하게 된다.

1. 론리연산자

우리는 일상적으로 흔히《옳다》,《틀린다》와 같은 말을 많이 한다. 생활에서 흔히 쓰는 이러한 말을 표현하는것이 콤퓨터에서 론리형자료이다. 론리형자료에서 《옳다》는 참(true), 《틀린다》는 거짓(false)으로 표현한다. 연산에는 수값들끼리 진행하는 산수연산도 있지만 《참》이나 《거짓》과 같은 론리자료로 진행하는 론리연산도 있다.

모든 연산에는 연산수와 연산자가 있다.

론리연산에서 연산수는 론리형자료이다. 론리연산자는 여러 조건들사이의 론리연산을 진행하는데 쓰이는 연산자이다. 론리연산자에는 론리적(And), 론리합(Or), 론리부정(Not)이 있다.

론리연산자와 괄호를 리용하여 련결하여놓은 식을 **론리식**이라고 한다.

1) 론리적(And)

두 연산수가 동시에 참(true)일 때에만 참을 주는 연산이다. 즉 두 연산수가운데서 하나라도 거짓(false)이면 거짓을 준다.

콤퓨터에서는 참을 1로, 거짓을 0으로 표시한다.

A	B	A and B
false	false	false
false	true	false
true	false	false
true	true	true

2) 론리합(Or)

두 연산수가운데서 어느 하나라도 참(true)이면 참(true)을 주는 연산이다. 두 연산수가 둘 다 거짓(false)인 경우에만 거짓을 준다.

A	B	A or B
false	false	false
false	true	true
true	false	true
true	true	true

3) 론리부정(Not)

연산수가 참(true)이면 거짓(false)을 주고 거짓(false)이면 참(true)을 주는 연산이다.

A	Not A
false	true
true	false

C언어에서는 이 론리연산자들을 어떻게 표시하는가?

여기서는 다음과 같은 연산자를 리용한다.

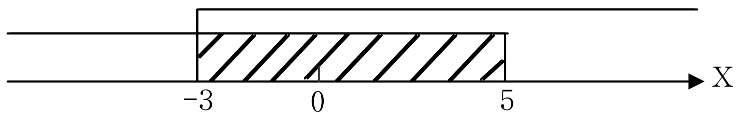
연산자	**의미**	**례**
&&	론리적	a&&b
||	론리합	a||b
!	론리부정	!a

론리적과 론리합은 두개의 연산수를 가지나 론리부정은 연산수를 하나밖에 가지지 않는다.

론리연산은 프로그람에서 비교식들사이의 결합연산에 리용한다.

례: 다음과 같이 수축에서 빗선친 구역을 론리식으로 표시해보아라.

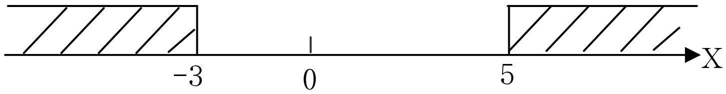
①



수학적표시: $x \geq -3 \wedge x \leq 5$

론리식표시: x>=-3&&x<=5

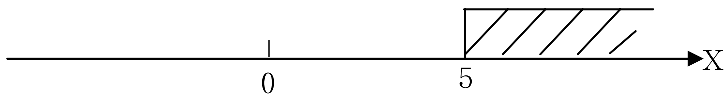
②



수학적표시: $x \leq -3 \vee x \geq 5$

론리식표시: x<=-3||x>=5

③



수학적표시: $x \geq 5$

론리식표시: x>=5 또는 !x<5

우의 식의 결과를 따져보자.

x=0이라고 하면 첫번째 그림에서 빗선친 구역(첫째 론리식)에 속하므로 참의 결과가 나와야 할것이다.

0>=-3 && 0<=5 = true

참 참

참



탐 구

다음 론리식의 결과는 무엇인가?

x>=-3&&x<=5||!x<5&&(x<=-3||x>=5)

2. 비트연산자

우리는 이미 여러가지 진수체계들에 대하여 학습하였다.

콤퓨터는 내부적으로 2진수(0,1)를 리용하는 기계이다. 콤퓨터에서 0과 1만을 기록할수 있는 한자리 기록위치를 비트(bit)라고 하였다.

비트연산은 두 연산수의 대응하는 비트들끼리 진행하는 연산이다.

비트연산은 콤퓨터의 장치내부에서 진행되는 연산들을 아는데서 매우 중요한 연산이라고 말할수 있다.

비트연산에는 비트론리연산과 비트밀기연산이 있다.

비트론리연산은 비트들사이에 진행되는 론리연산이다.

비트밀기연산은 비트단위로 오른쪽 또는 왼쪽밀기를 진행하는 연산이다.

1) 비트론리연산자

비트론리연산자에는 다음과 같은것들이 있다.

연산자	의미
&	비트론리적
|	비트론리합
ˆ	비트배타적론리합
~	비트론리부정

매개 비트론리연산의 기능은 이미 학습한 론리연산과 같은데 1을 참으로, 0을 거짓으로 생각하고 결과를 구하되 결과 역시 참을 1로 표시하고 거짓을 0으로 표시하면 된다. 여기서 비트배타적론리합은 비트단위의 두 연산수의 값이 서로 다를 때에만 론리합의 연산결과를 주고 같을 때에는 0의 결과를 주는 연산이다.

A	B	AˆB
0	0	0
0	1	1
1	0	1
1	1	0

례 1: $34_{10} \& 60_{10} = 0100010_2 \& 0111100_2 = 0100000_2 = 32_{10}$

$$\begin{array}{r} 0\ 1\ 0\ 0\ 0\ 1\ 0 \\ \&\quad 0\ 1\ 1\ 1\ 1\ 0\ 0 \\ \hline 0\ 1\ 0\ 0\ 0\ 0\ 0 \end{array}$$

$$34_{10} | 60_{10} = 0100010_2 | 0111100_2 = 0111110_2 = 62_{10}$$

$$\begin{array}{r} 0\ 1\ 0\ 0\ 0\ 1\ 0 \\ \underline{|\quad 0\ 1\ 1\ 1\ 1\ 0\ 0} \\ 0\ 1\ 1\ 1\ 1\ 1\ 0 \end{array}$$

$$34_{10} \hat{}\ 60_{10} = 0100010_2 \hat{}\ 0111100_2 = 0011110_2 = 30_{10}$$

$$\begin{array}{r} 0\ 1\ 0\ 0\ 0\ 1\ 0 \\ \underline{\hat{}\quad 0\ 1\ 1\ 1\ 1\ 0\ 0} \\ 0\ 0\ 1\ 1\ 1\ 1\ 0 \end{array}$$

$$\sim 34_{10} = \sim 0100010_2 = 1011101_2 = 93_{10}$$

$$\begin{array}{r} \underline{\sim\quad 0100010} \\ 1011101 \end{array}$$

비트론리연산자들이 들어있는 식을 쓸 때에는 복잡성을 피하기 위하여 비트론리식에 괄호를 쓰는것이 좋다.

비트론리연산자들은 옹근수형에 대하여서만 적용할수 있다.

2) 비트밀기연산자

비트밀기연산자에는 왼쪽밀기연산자(<<)와 오른쪽밀기연산자(>>)가 있다.

비트밀기연산자가 들어간 식의 일반형식:

연산수1<<연산수2 연산수1>>연산수2

여기서 첫 식은 연산수1을 연산수2의 회수만큼 왼쪽밀기하라는 식이고 둘째 식은 연산수1을 연산수2의 회수만큼 오른쪽밀기하라는 식이다.

례 2: $4_{10} << 2 = 00100_2 << 2 = 10000_2 = 16_{10}$

$16_{10} >> 2 = 10000_2 >> 2 = 00100_2 = 4_{10}$



탐 구

수를 오른쪽밀기할 때와 왼쪽밀기할 때의 결과는 본래의 수와 어떤 관계를 가지는가?

3. 조건명령문

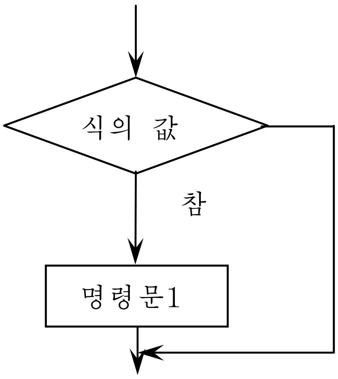
조건명령문은 조건에 따라 실행순위를 조종하는 명령문이다.
조건명령문에는 두갈래조건명령문과 여러갈래조건명령문이 있다.

1) 두갈래조건명령문

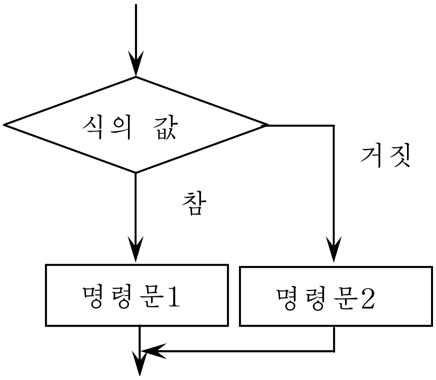
두갈래조건명령문은 조건에 따라서 두 갈래가운데서 어느 한 처리를 진행하는 명령문이다.

일반형식:

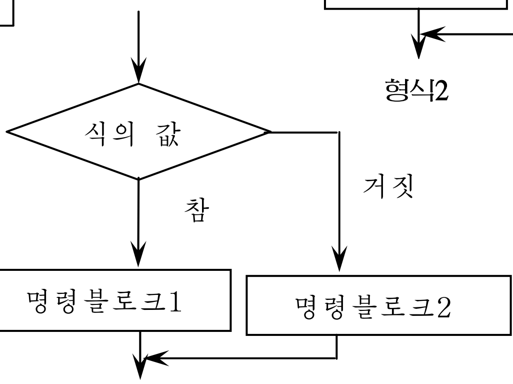
```
형식1: if (식) <명령문1>
형식2: if (식) <명령문1>
       else <명령문2>
형식1: if (식)
        {
            명령블로크1;
        }
       else
        {
            명령블로크2;
        }
```



형식1



형식2



형식3

기능:

· 형식1: 식의 값을 계산하여 식의 값이 참(0이 아니면)이면 명령문1을 실행한다.

· 형식2: 식의 값을 계산하여 식의 값이 참이면 명령문1을, 거짓이면 명령문2를 실행한다.

· 형식3: 식의 값을 계산하여 식의 값이 참이면 명령블로크1을, 거짓이면 명령블로크2를 실행한다.

우의 형식들에서 해당한 조건밑에 수행해야 할 명령문이 여러개 되는 경우 명령문들의 모임을 **명령블로크**라고 하였다.

명령블로크는 { }로 묶어 표시한다.

if문안에 또 if문을, else문안에 다시 if문을 쓸수 있다. 이런 경우 else는 제일 가까운 if문과 대응한다.

례 1: 수값을 입력하면 짝수인가 홀수인가를 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int a;
  scanf("%d", &a);
  if(a%2==0)  printf("even number\n");
  else printf("odd number\n");
}
```

례 2: 수값을 입력하여 정수이면 +기호를 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int a;
  scanf("%d", &a);
  if(a>0) printf("+\n");}
```

례 3: 두 수 a, b가운데서 큰 수를 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int a,b;
  scanf("%d%d", &a, &b);
  if(a>b) printf("big number=%d\n", a);
  else printf("big number=%d\n", b);
}
```

우리가 이미 절의 앞부분에서 설정한 과제에서 성적 mark가 4.5이상인가 아닌가에 대한 처리는 조건명령문을 리용하여 다음과 같이 쓸수 있다.

```
if(mark>=4.5)
    printf("최우등\n");
else
   {
     if(mark>=3.5)
       printf("우등\n");
     else
       printf("보통\n");
   }
```

2) 여러갈래조건명령문

여러갈래조건명령문은 식의 값에 따라 여러 갈래가운데서 어느 하나의 처리를 진행하는 명령문이다.

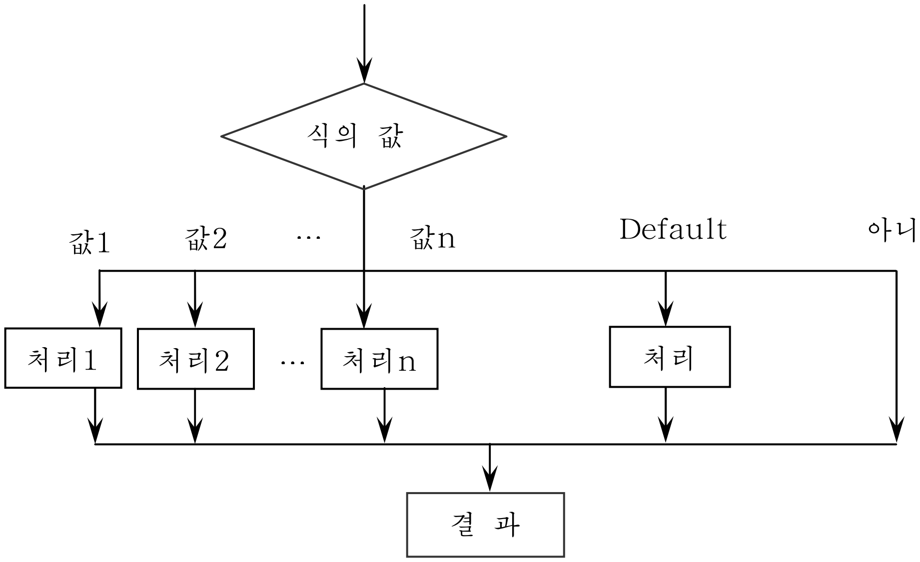
일반형식:

```
switch (식)
 {
case 값1:
        처리1;
         [break;]
case 값2:
        처리2;
         [break;]
        …
```

```
case 값n:
        처리n;
         [break;]
default:
        처리;
        }
```

기능:

식의 값을 판정하여 값1, 값2, …, 값n가운데서 식의 값과 일치한 값에 해당한 처리를 진행한다. 처리를 진행한 후에는 switch문에서 빠져나온다. break문을 만나면 즉시 switch문에서 빠져나온다. 만일 식의 값과 일치한 값이 없으면 default로 지정한 처리를 진행한다.



switch문에서 사용하는 식이나 값은 반드시 옹근수나 문자형과 같이 순서있는 자료여야 한다.

break문이 없는 경우에는 선택된 처리를 실행한 후에 다시 다음 case처리에 들어간다. 다시말하여 break문은 처리가 실행된 후에 그 switch문에서 빠져나오기 위하여 사용된다.

례 4: 두개의 옹근수를 입력하고 연산기호(한개의 기호)를 입력하면 두개의 수값에 대한 해당한 연산을 수행하여 결과를 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>
int main()
```

```
{ float a,b;
  char c;
  scanf("%f%f", &a, &b);
  scanf("%s", &c);
  switch(c)
  {
   case '+':
    printf("%f\n", a+b);
   case '-':
    printf("%f\n", a-b);
   case '*':
    printf("%f\n", a*b);
   case '/':
    printf("%f\n", a/b);
   default:
    printf("좋습니다.");
   }

}
```

례 5: 2010년 1월 1일이 금요일이라는것을 알고 이달의 임의의 날자를 입력하면 요일을 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  short day;
  scanf("%d", &day);
  switch(day mod 7)
  {
   case 1:
    printf("friday\n");
   case 2:
    printf("saturday\n");
   case 3:
    printf("sunday\n");
   case 4:
     printf("monday\n");
```

```
    case 5:
     printf("tuesday\n");
    case 6
     printf("wednesday\n");
    case 0:
     printf("thursday\n");
    }
}
```

3) 무조건이행명령문

프로그람집행과정에는 코드에 씌여진 순서대로가 아니라 필요에 따라 실행순위를 변화시켜야 할 때가 있다. 이렇게 순위를 변경시켜 어떤 명령문에로 뛰여넘게 하는 명령문이 바로 무조건이행명령문이다.

일반형식:

goto 표식;

기능:

표식이 붙은 명령문에로 이행한다.

이때 표식에는 수값을 쓸수 없다. 변수이름만들기규칙에서와 같이 문자로 시작하여 문자, 수자들의 결합으로 지어야 한다.

례 6: S=1+2+…+100을 계산하여라.

```
#include <stdio.h>
void main(void)
{
  int i=1,s=0;
  lp:
    s+=i;
    i++;
    if(i<=100) goto lp;
  printf("sum=%d\n",s);
}
```

C프로그람에서 goto문을 사용하면 프로그람의 구조가 복잡하게 되며 여러가지 오유의 발생원인으로 되므로 goto문의 사용은 될수록 금지하고있다.

우리가 이 절에서 학습한 내용을 종합하여 이미 절의 앞부분에서 설정하였던 과제의 코드를 쓰면 다음과 같다.

<코드>

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{float mark;
scanf("%f", &mark);
  if(mark>=4.5)
    printf("최우등\n");
  else
    {
     if(mark>=3.5)
      printf("우등\n");
     else
      printf("보통\n");
    }
}
```

해 설

이미 알고리듬에서 본바와 같이 성적 mark가 4.5이상인가를 판정하여 크면 《최우등》이라고 출력하며 아니면 다시 3.5이상인가 아닌가에 대한 판정을 진행한다. 3.5보다 크면 《우등》이라고 출력하며 아니면 《보통》이라고 출력한다.

련습문제

1. 다음 연산의 연산순위를 밝히고 연산결과를 구하여라.
   A=true, B=false, C=true
   1) A&&B||C 2) A||B&&C 3)!A&&C||B&&!C 4) A||B&&A||C
2. 다음 비트연산의 결과를 구하여라.
   1) 30&45 2) 25|12 3) 21^30 4) ~21&30|40
3. 다음 식의 결과를 구하여라.
   x=-3, y=6일 때
   1) x<0&&y>1 2) x>0||y<7 3) !(x<0)&&y<-5
   4) x<0||y>0 5) x<0&&x>-5 6) y>0&&y<10
4. 세 수 a, b, c가운데서 제일 큰 수를 출력하는 프로그람을 작성하여라.
5. 3각형의 세 변 a, b, c가 주어졌을 때 3각형을 이룰수 있는가를 판정하고 이룰수 있으면 면적을 구하여 출력하는 프로그람을 작성하여라.
6. 2차방정식의 풀이를 구하는 프로그람을 작성하여라.
7. 점 M(x, y)가 주어졌다. 이 점이 중심이 (0, 0)이고 반경이 R인 원안에 놓이는가를 판정하여 출력하는 프로그람을 작성하여라.

8. 수값이 10개 주어진다. 매 수값에 대하여 정수인가, 부수인가, 령인가를 판정하여 출력하는 프로그람을 작성하여라.

9. 10개의 수가 주어진다. 매 수가 짝수인가, 홀수인가를 판정하여 출력하는 프로그람을 작성하여라.

10. 세 점 A, B, C가 주어졌다. 이 세 점이 한 직선에 놓이는가를 판정하는 프로그람을 작성하여라.

11. y의 값을 구하는 프로그람을 작성하여라.

$$y=\begin{cases} x, & i=0 \\ \sin x, & i=1 \\ \cos x, & i=2 \end{cases}$$

12. y의 값을 구하는 프로그람을 작성하여라.

$$y=\begin{cases} x^2, & i=1 \\ x, & i=2, 3, 6, 7, 8 \\ \sin x, & i=9, 10, 11 \\ (\tan x)^2+1, & i>11 \end{cases}$$

13. if명령문과 switch문의 공통점과 차이점을 말하여라.

14. 세 수 a, b, c를 크기순서로 배렬하여라.

15. a값을 입력하여 왼쪽으로 1bit씩 밀기할 때마다 나오는 결과들을 출력하는 프로그람을 작성하여라.(5번만 밀기한다.)

16. 2010년 1월 1일이 금요일이라는것을 알고 2월의 임의의 날자를 입력하면 요일을 출력하는 프로그람을 작성하여라.

17. a와 b가 임의의 옹근수일 때 a^2+b^2의 값이 100보다 크면 100의 자리수, 100보다 작으면 10의 자리수를 출력하는 프로그람을 작성하여라.

18. 두 원의 중심점의 자리표와 반경이 각각 주어졌을 때 두 원이 사귀였는가를 판정하는 프로그람을 작성하여라.

19. 수 x가 주어졌을 때 20, 40, 60가운데서 가장 가까운 수값을 출력하는 프로그람을 작성하여라.

20. 기호를 입력한다. 기호가 A인것과 아닌것의 개수를 출력하는 프로그람을 작성하여라.



콤퓨터상식



《해커》란 말의 진짜 의미

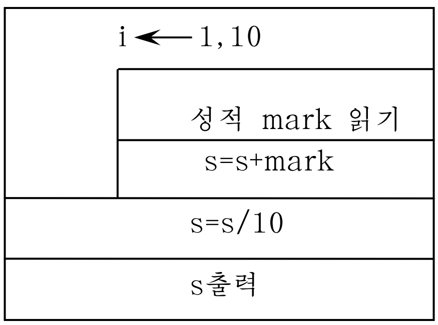
콤퓨터범죄가 성행하는 속에서 어느덧 사람들은 《해커(hacker)》라는 말을 《콤퓨터망에서 부정적인 일을 하는 사람을 통털어 이르는 말》처럼 리해하고있다. 그러나 이 말의 원래의 의미는 《손도끼 하나로 집을 짓는 사람》 혹은 《곡괭이로 땅을 파는 사람》이라는 뜻으로서 원래는 콤퓨터를 깊이 파고들어 많은 지식을 소유하고있으며 탐구적이고 창조적인 재능에 넘친 프로그람개발자를 존경하여 부르는 말이였다. 해커들이 콤퓨터망에서 다른 사람들의 체계에 침투하여 들어가는 경우에도 나쁜 의도를 가져서라기보다 순수한 호기심이나 탐구심으로부터 출발한 경우가 많았다. 해커란 말이 콤퓨터분야에서 쓰이기 시작한것은 죠세프 와이젠바움이라는 사람이《콤퓨터의 힘과 인간의 리성》이라는 책에서 탐구적인 프로그람개발자들을 이 말에 비유하여 표현한 때부터 생겨났다.

제6절. 10명 학생의 평균성적계산

이 절에서는 10명 학생의 평균성적을 계산하는 프로그람을 작성하는 과정을 통하여 순환명령문에 대하여 학습하기로 한다.

10명 학생의 평균성적을 계산하는 프로그람을 작성하자. 이 문제를 풀려면 10명 학생의 성적을 입력하여 더한 다음 10으로 나누어 평균값을 계산하여야 한다. 알고리듬은 다음과 같다.

<알고리듬>



알고리듬에서 보는바와 같이 이 문제는 i값을 1부터 10까지 1씩 증가시키면서 매번 증가된 i번호에 해당한 학생성적 mark를 입력하고 그것을 합을 나타내는 변수 s에 더하여야 하며 성적합계산이 끝나면 전체 성적의 합을 10으로 나누어 평균값을 계산하여야 한다.

이렇게 주기적인 반복과정이 들어있는 알고리듬은 순환명령문을 통하여 실현할수 있다.

1. 순환명령문 for

우리는 이미 앞절에서 조건명령문과 무조건이행명령문에 대하여 학습하였다.
아래의 프로그람을 보기로 하자.

```
main()
{ int i;
  l: i=i+1;
       s=s+i;
       if (i<10) goto l;
 }
```

이 프로그람에서 3, 4, 5행은 s=s+i를 구하기 위한 명령문들이다.

이 과정을 다시 보면 i=i+1은 1, 2, 3,…와 같이 1씩 증가되는 값들을 취해주는 명령문이며 if (i<10) goto l은 i의 끝값이 10이 되였는가 판단하며 아직 되지 않은 경우에는 i를 증가시키는 위치로 다시 순환시켜주기 위한 명령문이다. 즉 이 프로그람에서는 s=s+i를 위해 i를 1부터 1씩 증가, i<10인가 판단, l행에로 이행하는 과정이 필요하다. 이 세 명령의 기능을 대신하여 간단히 하나로 쓸수 있는 명령이 바로 for명령문이다.

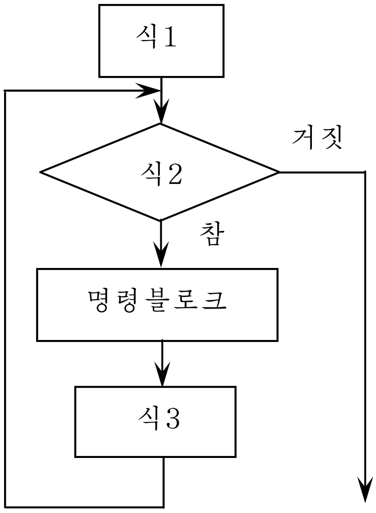
for명령문의 형식은 다음과 같다.

일반형식:

```
for ([식1];[식2];[식3])
 {
      명령블로크;
 }
```

기능:

① 식1평가
② 식2평가
③ 식2의 값이 참이면 명령블로크를 실행하고 ④에로 이행, 거짓이면 for다음 명령문에로 이행하여 순환을 끝낸다.
④ 식3을 평가하고 ②에로 이행



례 1: s=1+2+3+…+10을 계산하는 프로그람을 작성하여라.

```
#include <stdio.h>
void main(void)
{ int i, s=0;
   for(i=1;i<=10;i++)
       s+=i;
   printf("sum=%d\n",s);
}
```

이 프로그람에서 식 1은 i=1, 식 2는 i<=10, 식 3은 i++이다. 그러므로 i=1을 수행하여 i변수에 1을 넣고 i<=10인가 판정하여 참이면 s+=i 명령문을 수행한 다음 i++을 수행하여 i를 하나 증가시키고 다시 for명령의 i<=10판정에로 넘어간다. 만일 i<=10이 거짓이면 즉 i가 10을 넘었으면 for순환의 다음에로 넘어가 printf명령문을 실행함으로써 s를 출력한다.

for명령문에서 주의할 점은 다음과 같다.

· 식1은 순환부에서 한번만 실행된다. 그러므로 순환에서 변수에 대한 초기화에 리용되며 생략할수도 있다.

식 1에 여러 명령문을 쓸수도 있는데 이때에는 명령문들을 반점으로 구별하여 쓴다.

우의 례에서 식 1자리에 s=0도 쓸수 있는데 그렇게 하려면 다음과 같이 쓸수 있다.

```
for(s=0,i=1;i<=10;i++)
  s+=i;
```

· 식 2는 순환을 위한 조건이며 생략되면 무한순환이 된다. 그러므로 순환부에는 무한순환이 되지 않도록 조건식이나 break문을 리용하여 순환에서 빠질수 있는 명령문이 있어야 한다.

· 식 3은 순환을 진행할 때마다 실행되는 부분으로서 순환변수를 걸음값만큼씩 변화시키는데 리용된다.

생략하면 이 부분이 명령블로크안에 있을수도 있다.

3개의 항목이 다 생략될수도 있다.

```
for( ; ;)
 명령문;
```

· 명령문이 생략된 경우는 수행되는 명령문은 없지만 식 1, 식 2, 식 3에 대한 판단을 계속하면서 순환은 계속된다. 이것을 **빈순환**이라고 한다.

빈순환은 시간지연에 리용된다.

```
for (i=1;i<50000;i++);
```

· for순환에서 break문이나 continue문을 리용하면 무한순환을 피할수 있다.

break: 순환중지
continue: 순환계속

continue문이 실행되면 이 명령부터 순환부의 마지막명령문까지를 무시하고 다음순환에로 넘어간다.

break문이 실행되면 이 명령문 다음부터 순환부의 마지막명령문까지는 무시되고 순환이 중지된다.

례 2: s=2+4+6+…+500의 합이 1 000을 넘을 때 순환을 중지하고 합을 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int i, s=0;
  for(i=0;i<=500;i+=2)
   {
    s+=i;
    if(s>1000) break;
    }
  printf("s=%d\n", s);
}
```

례 3: 각 x가 -360° — +360° 까지 변할 때 매 각의 sin, cos, tan값을 출력하는 프로그람을 작성하여라.

이 문제에서는 각 x가 ±90° , ±270° 일 때에는 tan값이 정의되지 않으므로 이 값일 때는 피해야 하므로 continue문을 쓰는것이 적합하다.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
  int i;
  float s,c,t,x;
  for(i=0;i<=360;i++)
   {
    x=i*3.14/180;
    s=sin(x);
    c=cos(x);
    if(i==90 || i==270)
     { printf("%d: %f %f no \n", i,s,c);
       continue;
      }
    t=tan(x);
    printf("%d: %f %f %f\n", i,s,c,t);
   }
}
```

례 4: 2010년 1월 1일이 금요일이라는것을 알고 임의의 월, 일을 입력하면 요일을 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int i,m,n,s=0,d;
```

```
scanf("%d%d",&m,&n);
for(i=1;i<=m-1;i++)
 switch(i)
 {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
      s=s+31;
      break;
    case 2:
      s=s+28;
      break;
    case 4:
    case 6:
    case 9:
    case 11:
       s=s+30;
       break;
     }
  s=s+n;
  d=s%7;
  switch(d)
  {
   case 0:
    printf("목요일");
    break;
   case 1:
    printf("금요일");
    break;
   case 2:
    printf("토요일");
    break;
   case 3:
    printf("일요일");
    break;
```

```
     case 4:
      printf("월요일");
      break;
     case 5:
      printf("화요일");
      break;
     case 6:
      printf("수요일");
      break;
     }
}
```

순환명령문들은 어떤 명령문들을 하나로 묶어 이것을 반복하여 집행하도록 하는 명령문이다.

앞에서 설정한 문제에 대한 코드를 보자.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  float mark,s=0;
  int i;
  for(i=1;i<=10;i++)
  {
    scanf("%f",&mark);
    s=s+mark;
   };
   s=s/10;
   printf("%f\n", s);
}
```

해 설

이미 앞에서 설정된 문제에서 1부터 10까지 1씩 증가한다는 내용은 for문의 시작부분으로 대신하였고 이 조건하에서 반복하여야 할 입력과 성적합계산은 순환부안에 넣었다. 다음 순환부가 끝나면 성적을 10으로 나누어 출력하였다.

2. 순환명령문 while

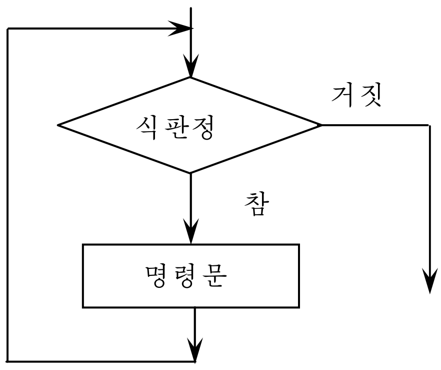
while은 《…하는 동안》이라는 뜻으로서 조건이 만족되는 동안 처리를 진행하게 하는데 리용된다. 앞에서 설정하였던 문제를 while문으로 작성하면 다음과 같다.

```
#include <stdio.h>
    main()
    { float s=0;
      int i=1;
      while(i<=10)
      {
       scanf("%f",&x);
       s+=x;
       i++;
       };
       printf("%f\n",s/10);
     }
```

여기서 보는것처럼 i는 인원수를 세는 변수로서 성적입력과 합구하기는 변수 i가 10이 아닌 동안 계속 반복해야 할 명령문들이다. 즉 while옆에는 명령문을 반복해야 할 조건식을 넣는다.

일반형식:

```
형식1: while(식)
            명령문;
형식2: while(식)
          {
            명령블로크;
          }
```



기능:

식의 값이 참(≠0)인 동안 명령문 또는 명령블로크를 반복하여 실행하며 거짓이면 순환부 다음 명령문에로 이행한다.

이 순환명령문에서 주의할 점은 다음과 같다.

・ 이 명령문들가운데는 식의 값을 변화시키는 명령문이 꼭 있어야 한다.

이런 명령문이 없으면 무한순환에 빠질수 있다.

례 1: ……

```
i=0;
while(i<=10)
  printf("**");
```

이 례에서는 i=0을 실행한 후 **를 출력하는데 언제가도 i가 10을 넘을수 없으므로 무한순환에 들어가게 된다. 즉

```
 …
i=0;
while(i<=10)
 {
   i++;
   printf("**");
  }
```

i가 1씩 증가하므로 i<=10은 언젠가는 거짓으로 되여버린다.

・ 식의 값이 처음부터 거짓이면 순환부는 한번도 실행되지 않는다.

례 2: s=1+2+3+…+10을 구하는 프로그람을 작성하여라.

```
main()
 {
   int i=1, s=0;
   while(i<=10)
    {
      s+=i;
      i++;
    };
  printf("%d\n", s);
 }
```

례 3: $s=1^2+2^2+3^2+\cdots$의 합이 100을 넘을 때 마디번호와 그 합을 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  int i=1, s=0;
```

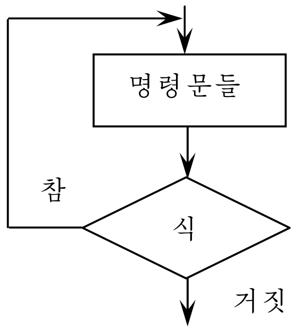
```
    while(s<=100)
     {s+=pow(i,2);
      i++;
     }
    printf("%d  %d\n",i, s);
  }
```

3. do~while문

이 명령문도 while문과 비슷한 명령문이다.
다만 조건식을 뒤에서 판정하게 되여있다.

일반형식:

```
do
 {
    명령블로크;
  }
while(식)
```



여기서 보는바와 같이 이 명령문은 우의 while문과 반대로 조건식을 마지막에 판정한다.

기능:

순환부를 먼저 실행하고 식의 값을 판정하여 이것이 참인 동안 순환부를 반복한다. 식의 값이 거짓으로 되면 while 다음명령문에로 이행하여 순환은 중지된다.

이 명령문에서는 순환부를 먼저 실행하고 조건식을 판정하므로 조건식이 참이든 거짓이든 순환부가 한번은 실행된다.

례 1: n^9(n은 1~50사이의 수)을 do～while명령문을 리용하여 구하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  int n,i=1;
  float y=1;
```

```
    scanf("%d", &n);
    if(1<=n && n<=50)
    {
     do
     {
       y*=n;
       i++;
      }
     while(i<=9)
     }
    printf("%f\n", y);
  }
```

례 2: a, b의 최대공통약수를 유클리드련제법으로 구하는 프로그람을 작성하여라.

```
  #include <stdio.h>
  #include <stdlib.h>

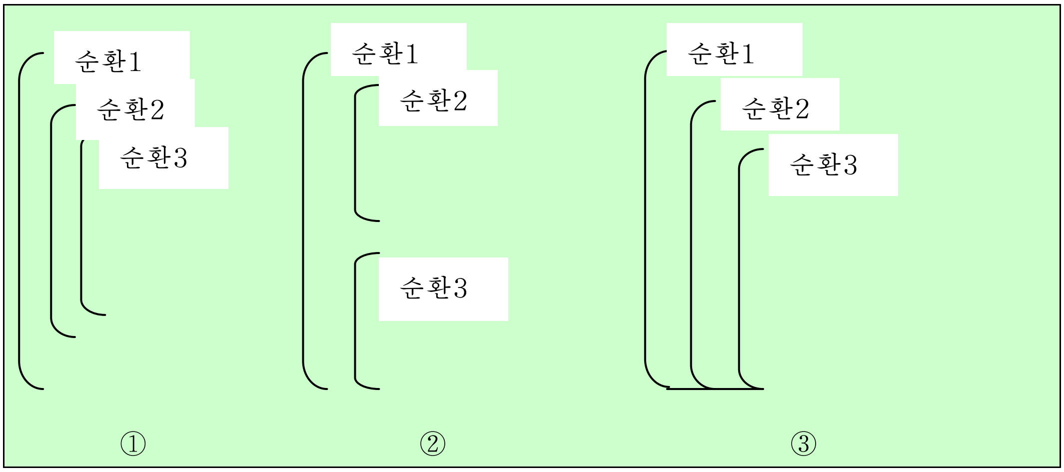
  int main(int argc, char *argv[])
  {
    int a,b,c;
    scanf("%d%d", &a, &b);
    do
    {
     c=a % b;
     a=b;
     b=c;
    }
    while(c!=0)
   printf("%d\n", a);
  }
```

이 문제는 while문으로 한다면 처음부터 c는 0이므로 순환부가 한번도 실행되지 않는다. 따라서 이 문제에서는 while문보다 do~while문이 더 합리적이라고 볼수 있다.

4. 다중순환

다중순환이란 순환부안에 또 다른 순환부를 가지는 순환명령문이다.

다중순환의 일반형태:



우에서 순환고리란 순환이 시작되여서부터 순환이 끝나는 곳까지의 위치를 말한다.

①형태레:

```
for(i=0; i<=10; i++)
 {
   for(j=0; j<=i; j++)
    {
      for(k=0; k<=j; k++)
         printf("%d%d%d\n", i, j, k);
      printf("%d%d\n", i, j);
    }
   printf("%d\n", i);
  }
```

②형태레:

```
for(i=0; i<=10; i++)
 {
```

```
        for(j=0; j<=10; j++)
          printf("%d\n", j);
        for(k=0; k<=10; k++)
          printf("%d\n", k);
        printf("%d\n", i);
      }
```

③형태례:

```
do
 {
   scanf("%d", &a);
   for(i=0; i<a; i++)
     for(j=0; j<=i+1; j+=2)
       printf("%d%d%d\n", a, i, j);
  }while(a!=0)
```

> **다중순환에서 순환변수들의 변화규칙**
> 다중순환에서 바깥순환변수가 한번 변할 때 안순환변수는 시작값부터 끝값까지 차례로 변한다.

이 규칙은 시계에서 초침이 60번 값이 변해야 분침이 한번, 분침이 60번 변한 다음에야 시침이 한번 변하는것과 비슷하다.

그러므로 시계의 시, 분, 초는 3중순환이며 다음과 같이 표시하면 시계의 시간을 모의할수 있다.

```
for(h=0; h<=11; h++)
 for(m=0; m<=59; m++)
   for(s=0; s<=59; s++)
    printf(%d%d%d\n", h, m, s);
```

례 1: 구구표를 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
  int i,j;
  for(i=1;i<=9;i++)
   for(j=1;j<=9;j++)
    printf("%d * %d = %d\n", i, j, i*j);

}
```

례 2: 1부터 50까지의 자연수들에 대하여 매 수들이 씨수인가 합성수인가를 판단하여 출력하는 프로그람을 작성하여라.

```
#include <stdio.h>
#include <stdlib.h>

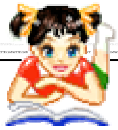
int main(int argc, char *argv[])
{
  int i,j;
  for(i=1; i<=50; i++)
   {
    for(j=2; j<=i-1; j++)
     {
      if(i%j==0) goto l;
      }
     printf("%d는 씨수이다.", i);continue;
   l:printf("%d는 합성수이다.", i);
     }
}
```

다중순환에서 주의할 점은 다음과 같다.

・ 다중순환에서 바깥순환은 안순환을 완전히 포함해야 하며 바깥순환고리와 안순환고리는 서로 사귀지 말아야 한다.

・ 바깥순환변수와 안순환변수는 서로 같을수 없다.

・ 안순환에서 바깥순환에로는 나갈수 있지만 바깥순환에서 안순환에로는 들어갈수 없다.



탐 구

1, 2, 3, 4 네 수를 리용하여 반복을 허용하면서 세자리수를 만들려면 몇중순환을 리용하여야 하는가?

련습문제

1. n!을 구하는 프로그람을 작성하여라.
2. 1/1*2+1/2*3+…+1/n*(n+1)을 구하는 프로그람을 작성하여라.
3. 1/1!-1/2!+1/3!-…+1/n!을 구하는 프로그람을 작성하여라.
4. $1+3+3^2+\cdots$의 합이 10 000을 넘을 때의 마디번호를 출력하는 프로그람을 작성하여라.
5. x의 n제곱을 표준함수를 리용하지 말고 구하는 프로그람을 작성하여라.(n은 옹근수)
6. 각 x가 -360° 부터 360° 까지 1° 간격으로 변할 때 삼각함수값들을 출력하는 프로그람을 작성하여라.
7. x가 1부터 100까지 0.1간격으로 변할 때 로그, 지수함수값을 출력하는 프로그람을 작성하여라.
8. 1+11+111+…+111…1(1의 개수 10개)을 구하는 프로그람을 작성하여라.
9. $1+4a+9a^2+\cdots$에서 n이 주어질 때 n번째 마디까지의 합을 구하는 프로그람을 작성하여라.
10. $1+3b+5b^2+\cdots$에서 n이 주어질 때 n번째 마디까지의 합을 구하는 프로그람을 작성하여라.
11. 첫째 마디가 n이고 공통비가 2인 같은비수렬에서 처음 몇개 마디의 합을 잡으면 그것이 1 000보다 크게 되겠는가를 출력하는 프로그람을 작성하여라.
12. 0, 1, 1, 2, 3, 5, 8,…의 첫 20번째 마디까지의 합과 20번째 마디를 출력하는 프로그람을 작성하여라.
13. 100개의 실험측정자료가 있다. 정수이면서 홀수번째인 자료들의 합과 개수를 출력하여라.
14. 1 111 — 5 000까지의 수들가운데서 10의 자리수자와 1의 자리수자가 같은 수들을 모두 출력하고 그 개수를 출력하여라.
15. a부터 b까지의 자연수들의 합을 출력하는 프로그람을 작성하여라.
16. 자연수가 주어졌다. 이 수가 씨수인가 합성수인가를 판정하여 출력하는 프로그람을 작성하여라.

17. 자연수가 주어졌다. 이 수가 합성수이면 이 수를 씨인수분해하는 프로그람을 작성하여라.
18. 두 수의 최대공통약수와 최소공통배수를 출력하는 프로그람을 작성하여라.
19. 두 수 a, b가 주어졌다. 이 두 수를 최대공통약수, 최소공통배수로 하는 두 수를 구하는 프로그람을 작성하여라.
20. 10부터 100까지 자연수가운데 수자 3이 모두 몇개 있는가를 구하는 프로그람을 작성하여라.
21. (2), (4, 6), (8, 10, 12), …인 수묶음이 주어졌다. 20번째 묶음까지 매 묶음의 합과 그 묶음을 출력하는 프로그람을 작성하여라.
22. (1), (2, 3, 4), (5, 6, 7, 8, 9), (10, 11, 12, 13, 14, 15, 16), …인 수묶음이 있다. 20번째 묶음까지의 매 묶음과 그의 합을 출력하는 프로그람을 작성하여라.
23. $s=1+x/1!+x^2/2!+x^3/3!+\cdots$을 소수점아래 6자리까지의 정확도로 구하는 프로그람을 작성하여라.
24. sinx의 값을 표준함수를 리용하지 말고 전개공식을 리용하여 소수점아래 5자리까지 정확도로 구하는 프로그람을 작성하여라.
25. ab*ba=403이 되는 a와 b를 구하는 프로그람을 작성하여라.
26. $s=1+21+321+1321+21321+321321+\cdots+\underbrace{321321\cdots321}_{\text{15자리}}$을 구하는 프로그람을 작성하여라.

27. $s=1+12+123+1231+12312+123123+\cdots+\underbrace{123123\cdots123}_{\text{15자리}}$을 구하는 프로그람을 작성하여라.

28. 11로 완제되면서 하나의 자리수자는 7인 6자리수가 있다. 이 수의 하나의 자리수자, 열의 자리수자, 백의 자리수자를 지워버리면 597이 된다. 이런 수는 몇개이며 어떤 수인가?
29. $s=1-12+123-1231+12312-123123+\cdots-\underbrace{123123\cdots123}_{\text{30자리}}$을 구하는 프로그람을 작성하여라.

30. 옹근수 n이 주어졌다. n의 약수들의 합과 그 개수를 출력하는 프로그람을 작성하여라.



콤퓨터상식



전자우편이란 어떤것인가?

현재 세계적으로 콤퓨터망을 통한 우편거래방식 즉 전자우편(E-mail)방식이 널리 리용되고있다. 어떤 사람들은 전자우편과 아주 친숙해져 친절하게 《누이동생》이라고 부르기까지 하고있다.

그러면 전자우편이란 무엇이며 여기에도 우편국과 우편함이 있는가?

통속적으로 일반우편과 달리 콤퓨터망을 통하여 주고받는 우편을 전자우편이라고 하며 이것을 실현할수 있는 체계를 전자우편체계라고 한다. 물론 전자우편체계에도 우편국과 우편함이 있다.

일반적으로 우리들은 콤퓨터망봉사제공자에 망가입신청을 하는데 이때 콤퓨터망봉사제공자는 필요한 수속처리와 함께 동시에 가입자에게 하나의 전자우편주소를 제공하게 된다. 그러면 이 콤퓨터망봉사제공자의 전자우편봉사기가 바로 《전자우편국》으로 되는데 이 우편국에서는 동시에 자기의 기억기안의 일정한 구역을 내여 하나의 《전자우편함》을 설정해놓는다. 이렇게 전자우편국과 전자우편함들은 콤퓨터망봉사제공자의 봉사기에 존재하게 되며 이것에 의하여 관리된다.

전자우편의 전송과정은 일반우편방식에서와 류사하다.

김동무가 안동무에게 전자우편을 보낼 때 전자우편은 먼저 자기의 전자우편봉사기에 보내여지며 곧 자기의 전자우편함에 넣어진다. 다음 전자우편봉사기는 김동무의 전자우편을 콤퓨터망을 통하여 안동무가 등록되여있는 전자우편봉사기에 전송하며 이것은 다시 안동무의 전자우편함에 넣어진다. 이렇게 되면 안동무가 전자우편쏘프트웨어를 리용하여 자기의 전자우편함을 조사해볼 때 김동무가 보낸 전자우편을 발견하고 열어보게 된다.

전자우편체계에서는 일반우편에서와는 달리 우편통과 우편함이 따로 설치되여있지 않다. 전자우편국은 곧 《우편구역》안의 매 사람들에게 배정된 우편함으로서 매 사람들은 보내는 우편을 자기의 우편함에 넣으면 된다. 이렇게 전자우편함이 전자우편국안에 직접 설치되여있기때문에 편지를 보내고 받는것이 모두 아주 빠르게, 아주 헐하게, 편리하게 진행되게 된다.

콤퓨터(제1중학교 제5학년용)

집 필	정금진, 리은경, 김창혁, 리영걸	**심 사**	심의위원회
편 집	길명희		
장 정	김순영	**교 정**	오혜란
낸 곳	교육도서출판사	**인쇄소**	평양고등교육도서인쇄공장
인 쇄	주체99(2010)년 11월 20일	**발 행**	주체99(2010)년 11월 30일

교-10—657 값 10 원